CENTRO UNIVERSITÁRIO FEI

PEDRO HENRIQUE DA SILVA MOURA

VINÍCIUS CALIL FERRAZ

DETECÇÃO DE CRIANÇAS EM IMAGENS PARA CONTROLE PARENTAL

São Bernardo do Campo

2020

PEDRO HENRIQUE DA SILVA MOURA
VINÍCIUS CALIL FERRAZ

DETECÇÃO DE CRIANÇAS EM IMAGENS PARA CONTROLE PARENTAL

Trabalho de Conclusão de Curso apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Ciência da Computação. Orientado pelo Prof. Dr. Rodrigo Filev Maia e coorientado pelo Prof. Dr. Guilherme Alberto Wachs Lopes.

São Bernardo do Campo

2020

Ficha catalográfica

Folha de aprovação

AGRADECIMENTOS

Agradecemos aos nossos familiares e colegas, que nos ajudaram, apoiaram em toda essa etapa. Para todos os professores dessa instituição que nos ensinaram tanto e principalmente aos professores Rodrigo Filev e Guilherme Wachs que são o orientador e co-orientador respectivamente.

RESUMO

Com o aumento do número de menores de idade utilizando a internet, surgem preocupações sobre a segurança dessas crianças nesse ambiente. Um exemplo que vem sendo debatido é a possibilidade de uma criança se passar por adulto em sistemas que não requerem identificação do usuário, tais como os *Marketplaces*. Como uma forma de mitigar tais situações, algumas companhias vêm se utilizando de visão computacional para identificação em tempo real do usuário por meio de webcam ou outros dispositivos. Nesse trabalho, é proposto um modelo computacional para identificação de crianças a partir de uma imagem (seja ela obtida em tempo real ou não) através da utilização de duas tecnologias: *Haar-cascade* e *Deep Learning*; uma para identificar rostos nas imagens e a outra para detectar se aquele rosto encontrado é de uma criança. Os resultados mostraram cerca de 73% de acerto médio nas imagens.

Palavras-chave: Face de crianças. Reconhecimento facial. Rede Convolucional. Haar Cascade. Processamento de imagem. Deep Learning. Machine Learning.

ABSTRACT

With the increase in the number of minors using the Internet, concerns about the safety of these children arise in this environment. An example that has been debated is the possibility of a child passing through an adult in systems that do not require user identification, such as the Marketplaces. As a way to mitigate such situations, some companies are using computer vision for real-time identification of the user through webcam or other devices. In this work, a computational model is proposed for the identification of children from an image (whether it is obtained in real time or not) through the use of two technologies: Haar-cascade and Deep Learning; one to identify faces in the images and the other to detect if that face found is of a child. The results showed about 73% of average accuracy in the images.

Keywords: Convolutional Network. Haar Cascade. Face of children. Facial recognition. Image Processing. Deep Learning. Machine Learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, a internet tem crescido de maneira exponencial. Um dos principais motivos por essa expansão é o acesso à tecnologia e aos computadores pessoais. Além disso, a tecnologia móvel tem levado a conectividade a diversos lugares e pessoas.

Dentre os principais público-alvo dessa expansão é o público jovem. Atualmente, no Brasil, 33% da população é criança (0 até 12 anos incompletos) e, dessa parcela, 49% das crianças de 9 a 10 anos utilizam o celular e 70% o computador, e de 11 a 12 anos 77% utilizam o celular e 64% o computador (MARQUES, 2014). Assim, é fácil observar que menores de idade são facilmente expostos aos riscos oferecidos pela internet e tecnologia em geral.

Pode-se separar o problema de exposição à internet em duas categorias: falta de controle de identificação do usuário; e exposição indevida do usuário. A primeira delas é muito recorrente em sistemas do tipo *Marketplace*, que conectam empresas fornecedoras a clientes através de um *website*.

Essa falta de controle para identificação do consumidor pode ser muito perigosa dependendo do produto que está sendo vendido. Por exemplo, certos produtos químicos (como alcóol isopropílico) não podem ser acessados diretamente por qualquer público. Contudo, é fácil para um menor de idade obter tais produtos, uma vez que não existe qualquer forma de certificação de que o consumidor seja de fato habilitado para a compra.

A segunda categoria de problemas está relacionada com a exposição do usuário na rede mundial. O crescimento das redes sociais (Facebook, Whatsapp, Instagram, Snapchat, Youtube, entre outras) têm facilitado a forma de se comunicar. Contudo, os dispositivos no qual fazem acesso a tais redes sociais possuem câmeras e mídias que podem ser facilmente expostas para o mundo. Como mostra MARQUES (2014), 79% das crianças e adolescentes que usam a internet possuem um perfil nas redes sociais.

Considerando a facilidade de acesso e publicação de mídias, a divulgação de materiais impróprios pode ser frequente. Esse é o caso onde surge o contexto de pornografia infantil. Esse assunto tem sido abordado em muitos fóruns de segurança e ainda é um grande problema na sociedade.

Essas duas categorias abordadas de problemas podem ocasionar traumas físicos e/ou psicológicos em menores de idade. Contudo, ainda há muita pesquisa que deve ser feita para restringir e proteger os menores de idade na internet.

Uma vez que a maioria dos dispositivos com acesso à internet apresenta câmera, é possível fazer o uso dessa tecnologia para identificar o usuário e, até mesmo, o conteúdo de infor-

mações que estão sendo postadas na rede. Assim, a proposta desse trabalho é criar um modelo computacional para identificação de menores de idade através de imagens.

1.1 OBJETIVO

Este trabalho tem por objetivo modelar e implementar um modelo computacional capaz de detectar menores de idade a partir de imagens digitais contendo faces frontais de pessoas.

1.2 MOTIVAÇÃO

Hoje a polícia utiliza um processo manual para identificar as imagens com o conteúdo de pornografia infantil, esse processo requer muito tempo e esforço, além de necessitar de auxílio de psicólogos. Com uma base muito grande existem muitas fotos que não são nem ao menos do tema de pornografia, com essas ferramentas a base ficará muito mais limpa, facilitando e diminuindo o trabalho dos oficiais.

2 TRABALHOS RELACIONADOS

Esse capítulo apresenta os trabalhos relacionados ao tema de identificação de menores de idade.

Muitos trabalhos têm focado na detecção de crianças em imagens. São diversas aplicações onde tais modelos podem ser aplicados. Em VitorinoAPR18, os autores propõem um modelo de detecção automática de pornografia infantil a partir de imagens. A principal motivação por detrás do trabalho é que um método automático de detecção pode diminuir consideravelmente os crimes cibernéticos com crianças e adolescentes. O modelo consiste de uma rede neural convolucional multi-camadas para tal detecção. Os autores afirmam que esse é um dos primeiros trabalhos que trouxe a aplicação de redes neurais convolucionais ao domínio de detecção de crianças.

Há trabalhos que são parecidos com a detecção de pornografia infantil, como apenas detectar pornografia em imagens. Em jin2018pornographic, foi proposto treinar um modelo de reconhecimento que não se especializa em uma parte do corpo e sim em padrões visuais discriminativos, ou seja, as imagens foram modeladas por regiões e utilizando múltiplas instâncias de treinamento para gerar um modelo genérico de reconhecimento. A intenção do trabalho era prevenir crianças de pesquisar sobre pornografia e ter acesso à imagens e também adultos que não deveriam ser expostos a esse conteúdo em determinados horários, como em horário de trabalho. Como resultado deste trabalho, foi possível gerar uma variedade de regiões com diferentes níveis de pornografia, podendo quantificar o quanto a região da imagem é pornográfica.

Há também trabalhos que utilizam técnicas parecidas como visto para detecção de pornografia e pedofilia online para detectar outro tipo de conteúdo, como por exemplo detectar violência em imagens. A ideia em brunomalveirapeixoto2018 era descrever melhor a ideia de violência para uma rede neural convolucional e isto poderia ser aplicado para proteger crianças de acessar conteúdo inapropriado e ajudar pais tomarem uma decisão melhor informada sobre o que seus filhos devem assistir. Neste trabalho, o aprendizado da CNN foi divido em objetivos como reconhecer lutas, explosões, sangue, etc. Para então fundi-los em uma meta-classificação e descrever a violência, resultando na possibilidade de classificar a violência podendo definir qual a faixa etária seria recomendada para assistir a cena.

Um assunto muito pesquisado na visão computacional e que que se encaixa no quadro de detecção de objetos em imagens é a detecção de sorrisos. Em (XIA; HUANG; WANG, 2017) o objetivo era destacar o impacto da discrepância entre o sorriso das crianças e adultos de maneira quantitativa para poder melhorar a solução para o problema de detecção de sorriso, pois

há diferença na estrutura facial de uma criança ou bebê para um adulto. Utilizando a estrutura da CNN e métodos de transferência de aprendizagem, como *Deep Adaptation Networks* (DAN) e *Joint Adaptation Network* (JAN), fez a CNN aprender o padrão de um sorriso de um adulto e adaptá-lo com sucesso para sorrisos de crianças.

Além da detecção e reconhecimento de objetos, em qawaqneh2017deep mostrou a CNN utilização da CNN para estimar idade das pessoas a partir de imagens faciais. Este trabalho expande o uso da CNN, pois é utilizada para reconhecimento facial, classificação de imagens e reconhecimento de objetos e neste trabalho a intenção era estimar idade das pessoas. O desenvolvimento deste trabalho utilizou uma CNN pré-treinada *VGG-Face* para reconhecimento facial como base, mantendo suas camadas convolucionais inalteradas, apenas substituindo as camadas totalmente conectadas por quatro novas camadas totalmente conectadas para treinar uma nova CNN capaz de estimar a idade das imagens faciais, o resultado é que o modelo proposto supera os trabalhos anteriores em 9% no *Audience database*, que possui imagens com muito ruído como baixa resolução, borrões, expressões faciais e iluminação pobre.

Ainda falando sobre estimar idade, há trabalhos que documentaram uma avaliação dos serviços de previsão de idade existentes, realizando uma análise avaliativa e comparativa dos serviços para identificar tendências e questões inerentes à seu desempenho como visto em (ANDA et al., 2018). A motivação deste trabalho estava em ajudar investigações policiais, pois em certas circunstâncias, a idade da vítima pode resultar na determinação da categorização de um crime, como abuso infantil. Realizando uma avaliação empírica baseada na observação com objetivo de encontrar o mínimo erro absoluto médio dentro dos diferentes serviços biométricos resultou que a quantidade de imagens marcadas com precisão com a idade e gênero disponível para pesquisadores é limitado. Portanto foi criado um gerador de conjunto de dados aleatório e balanceado para superar esse obstáculo combinando conjuntos de dados existentes. Além disso, foi incluído na coleção de imagens digitais, menores de idade, o que ajudou a avaliar razoavelmente o desempenho em relação aos homens e mulheres.

Outro uso interessante da CNN é o reconhecimento de gênero e emoções. Como pode ser visto em dehghan2017dager, foi desenvolvido um sistema *end-to-end* capaz de estimar os atributos faciais, incluindo idade, sexo e emoção, com baixas taxas de erro. A motivação deste trabalho estava em melhorar aplicações relacionadas ao controle de segurança, identificação de pessoas e interação humano-computador. Com 3 bases diferentes, foi realizado o treinamento através de redes neurais convolucionais, sendo que cada base foi utilizada para um objetivo diferente como reconhecimento de gênero, reconhecimento de emoções e estimar a idade. Cons-

truindo assim uma arquitetura que juntamente com os dados coletados internamente, pôde superar os algoritmos competitivos comerciais e acadêmicos em vários *benchmarks*. Pois obteve 90,31% de precisão para reconhecimento de gênero, 76,1% para reconhecimento de emoções e 70,5% para estimar a idade.

Outro trabalho com a motivação voltada para segurança é como o (TRAN et al., 2016), que criou uma estrutura chamada *Privacy-CNH* que utiliza um modelo de aprendizado profundo para detectar risco de privacidade em fotos, como por exemplo, imagens que contém dados sensíveis de pessoas. A intenção deste trabalho está em educar e proteger usuários, pois imagens com dados sensíveis podem ser usadas por criminosos cibernéticos para atividades fraudulentas, como o roubo de identidade. Foi utilizado neste trabalho a CNN para detectar objetos e aprender a diferença entre imagens que podem ser publicadas e imagens que devem permanecer em sigilo e também utilizar a transferência de aprendizado para aproveitar o aprendizado de outros conjuntos de dados. O método proposto é capaz de detectar automaticamente a privacidade em fotos de risco com a precisão de 95%, sendo que houve uma melhora na precisão entre 4% e 15% com a aprendizagem de transferência.

Em 2015 a CNN era explorada com propósito de detecção facial, como em (LI et al., 2015), a proposta era construir um classificador binário mais preciso, propondo assim uma melhoria ao algoritmo de Viola-Jones, que utiliza características tipo Haar para identificar o rosto em uma imagem, mas que é relativamente fraco em ambiente onde os rostos estão em poses variadas e expressões sob iluminação inesperada. A implementação mostra uma cascata de CNNs, sendo 3 CNNs para classificação binária se há ou não um rosto na imagem e 3 CNNs para calibrar o *frame* do rosto, uma vez que este método detecta padrões de um rosto de perfil e frontal. O resultado deste trabalho mostrou uma detecção mais rápida por conta da cascata de redes, no *benchmark* de detecção de face pública FDDB, o detector proposto superou os métodos da época, pois alcançou14 FPS para imagens VGA típicas na CPU podendo ser acelerado para 100 FPS na GPU.

Como foi citado o método Haar no parágrafo anterior, há pesquisas que focaram em avaliar classificadores faciais que utilizam o método Haar Cascade, pontuando cada característica facial contida na imagem que foi localizada, como vemos em padilla2012evaluation. O motivo de realizar essa avaliação é ajudar pesquisadores a escolher o melhor classificador para sua necessidade. Primeiro, foi identificado que a diferença entre os classificadores está no número de etapas e pelo tamanho mínimo dos rostos que podem ser detectados, então para medir a precisão desses classificadores foram utilizados 2 critérios. O primeiro critério utilizado

foi a pontuação, de 0 a 31, por cada característica identificada. Neste critério 100% das faces obtiveram pontuações iguais ou superiores a 27 para os classificadores utilizados. O segundo critério determinou se a região da face detectada é precisa verificando se as imagens tinham rostos detectados em uma imagem onde a altura e largura era menor que quatro vezes a distância dos olhos e o resultado foi que dos classificadores avaliados,100% das imagens tiveram rostos detectados.

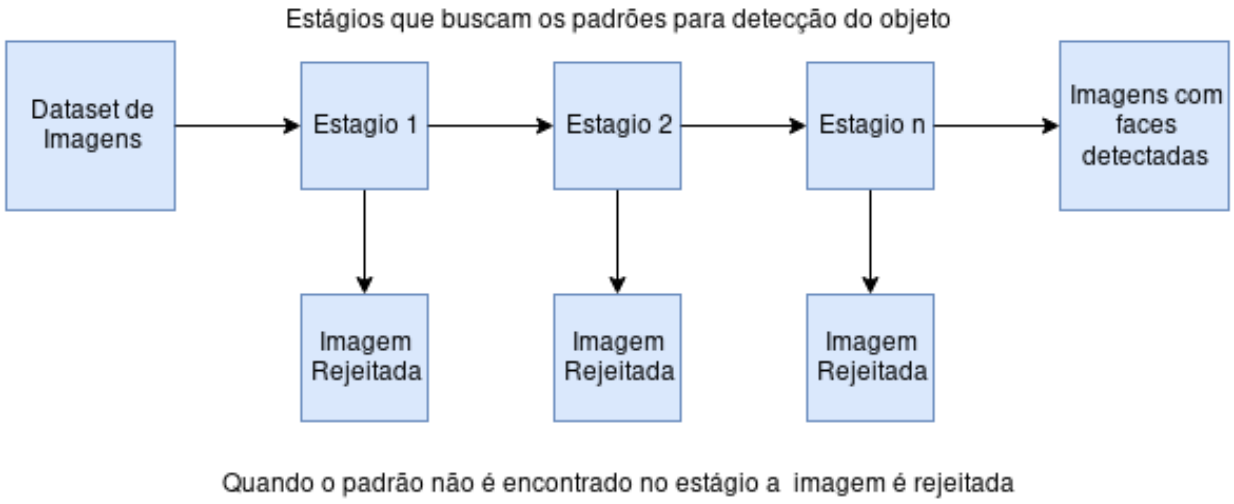
3 CONCEITOS FUNDAMENTAIS

Este trabalho tem como objetivo detectar a presença de crianças e adolescentes em imagens, motivado pela forma que a polícia faz para combater a pornografia infantil. Importante enfatizar que, para facilitar a compreensão, este texto refere-se à crianças e adolescentes até 17 anos de idade quando menciona a palavra "criança".

Para atingir o objetivo final é necessário que o sistema desenvolvido, primeiramente, detecte pessoas em uma imagem e em seguida classifique se a pessoa detectada é uma criança ou não.

Partindo desta ideia, para a detecção de uma pessoa, será implementado um algoritmo que é utilizado para detecção e reconhecimento de objetos chamado *Haar-Cascade*, que utiliza alguns estágios para detectar o objeto procurado na imagem. Para este trabalho, as pessoas serão detectadas na imagem apenas por suas características faciais. Veja na figura 1 o funcionamento do *Haar-Cascade*:

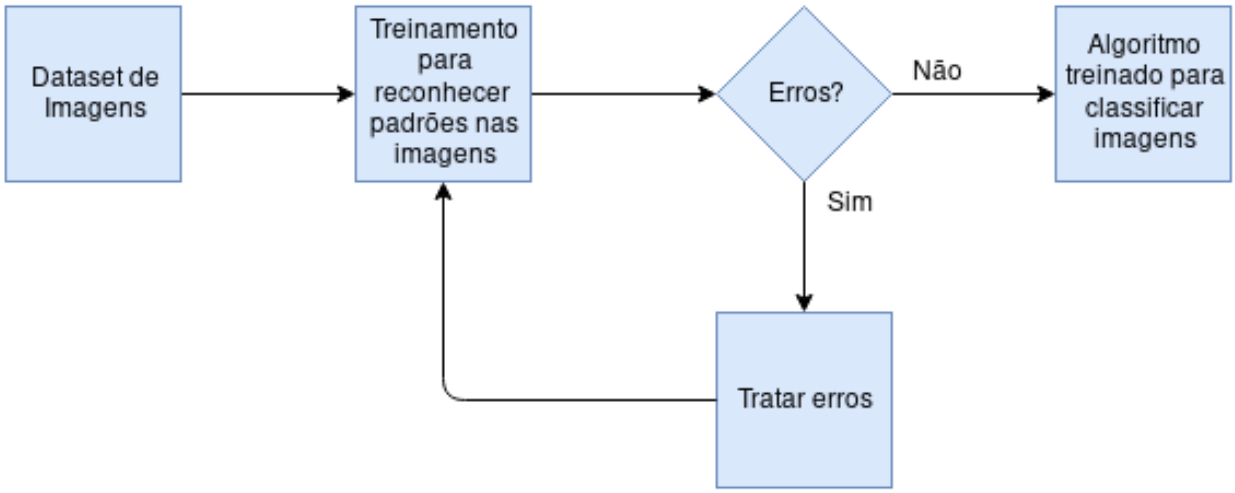
Figura 1 – Funcionamento do *Haar-Cascade* para detecção facial



Fonte: Os autores

Para o segundo passo, o algoritmo utilizado para obter a classificação da face detectada pelo *Haar-Cascade* entre criança e não-criança é a rede neural convolucional, também chamada de "CNN"(*Convolutional Neural Network*). A CNN, deve ser treinada para ser capaz de classificar se a figura em uma imagem é ou não uma criança. A figura 2 ilustra como esta classificação é feita:

Figura 2 – Passos para obter o algoritmo de classificação



Fonte: Os autores

Mais detalhes sobre a implementação são abordados no capítulo: "Metodologia". Neste capítulo será explicado os conceitos teóricos de ambos algoritmos e tecnologias utilizadas neste trabalho.

3.1 *MACHINE LEARNING*

Atualmente, muito se discute sobre o aprendizado de máquina, *Deep Learning* e inteligência artificial, e acredita-se que esses três termos são sinônimos, mas na verdade não é bem assim. Inteligência artificial é um conceito amplo que inclui o aprendizado de máquina ou *Machine Learning* como um dos seus recursos, e o *Deep Learning* é uma subcategoria do *Machine Learning*.

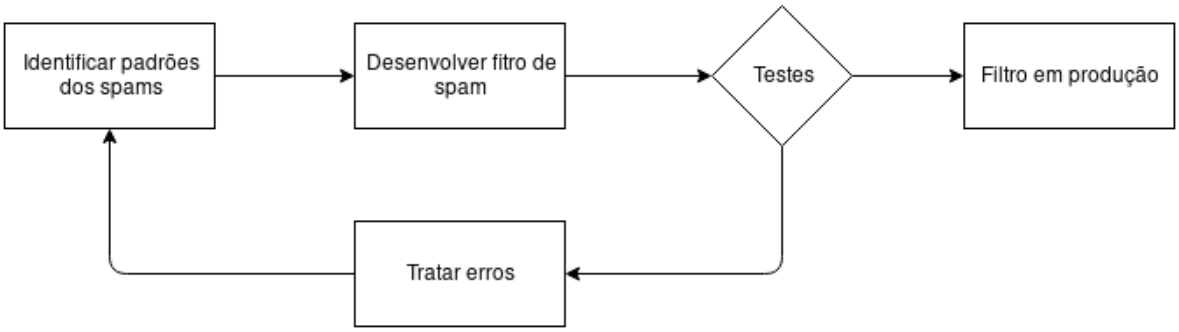
Machine Learning é uma área da computação que estuda a implementação de algoritmos que aprendem a executar uma tarefa baseando-se nos dados disponibilizados. Primeiramente, ao utilizar o *Machine Learning*, o algoritmo utilizado cria um modelo matemático a partir das informações extraídas dos dados e então utiliza esse modelo para classificar novos dados. Existem diversos algoritmos capazes de realizar isto, mas atualmente o que mais se utiliza são as redes neurais.

Para melhor compreensão de onde é possível aplicar *Machine Learning*, será utilizado como exemplo um filtro de *spam*, assim como feito em (GÉRON, 2017). Tradicionalmente, pode-se escrever um filtro de *spam* criando regras para identificar padrões que são comuns em

e-mails de *spam*, como palavras específicas presentes no corpo, assunto e até mesmo no campo de remetente. Após a criação do filtro, geralmente realiza-se alguns testes para verificar se o funcionamento está ocorrendo conforme o esperado e, caso não esteja, os erros devem ser analisados e as regras reescritas.

Veja que o parágrafo acima mostra um ciclo de um trabalho manual para que o filtro de *spam* possa entrar em produção. Porém, mesmo com o filtro funcionando corretamente e em produção, com o tempo, podem surgir novos padrões de *spam* e com isso o ciclo deve se repetir, desde identificar os padrões até a execução dos testes.

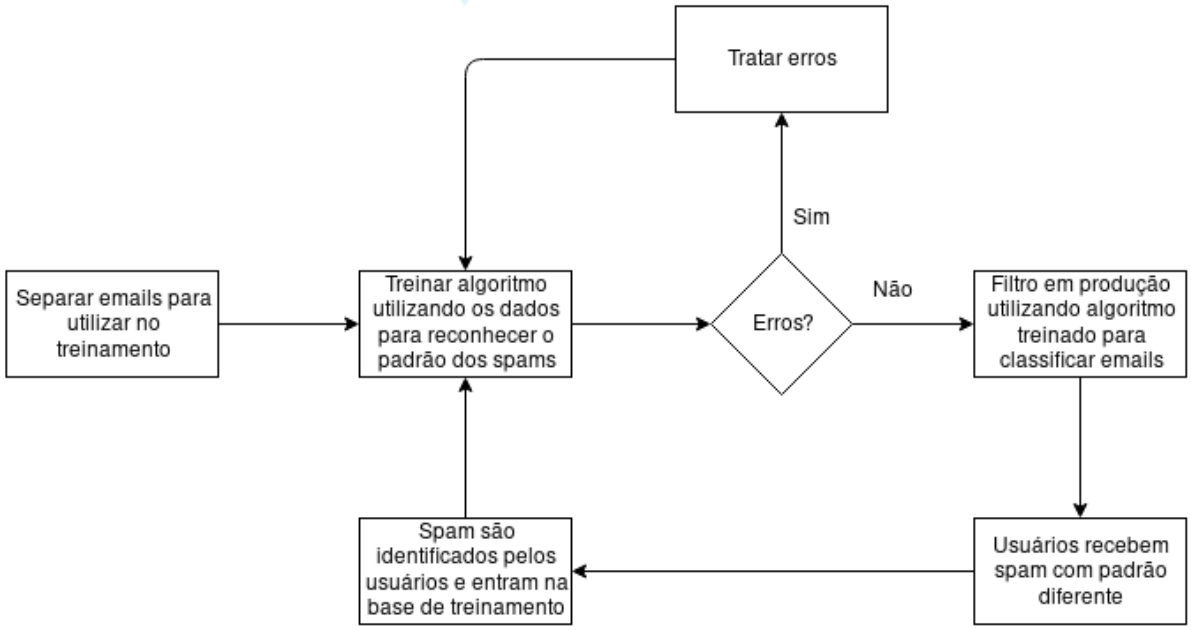
Figura 3 – Ciclo manual para identificar *spams*



Fonte: Os autores

Com este cenário, temos um trabalho manual que pode ser automatizado e ter uma eficiência melhor se utilizado o *Machine Learning*, como mostra a figura 4.

Figura 4 – Ciclo automatizado para identificar *spams*



Fonte: Os autores

3.1.1 Tipos de *Machine Learning*

Há alguns tipos de sistemas de *Machine Learning* que são categorizados em tipo de treinamento, tipo de aprendizado e tipo de funcionamento. Essas categorias podem ser combinadas de acordo com a necessidade. São elas:

a) Treinamento supervisionado, não supervisionado, semi-supervisionado ou aprendizado por reforço;
b) Aprendizado incremental, conhecido como *Learning-on-the-fly*, ou aprendizado por lote, chamado também de *batch learning*;
c) Funcionamento comparando dados conhecidos com novos dados ou detectar padrões nos dados conhecidos para criar um modelo preditivo.

Como supracitado, há quatro maneiras de realizar o aprendizado com *Machine Learning*. A definição principal no treinamento supervisionado é "classificação": rotula-se os dados antes do treinamento, criando classes para estes e assim garantindo a possibilidade de classificar novos dados. Por exemplo, se o objetivo de um trabalho é reconhecer tumores cerebrais em imagens de radiografia, antes do treinamento deve-se rotular as imagens que possuem e não possuem câncer, respectivamente como "positivo"e "negativo"e então com essas duas classes

de dados, utilizando um algoritmo correto para lidar com imagens e realizar o treinamento supervisionado, é possível obter a classificação da presença de tumor cerebral em outras imagens.

O treinamento não-supervisionado é exatamente o oposto do supervisionado, pois não se informa para o algoritmo o que são os dados e assim fica como dever da máquina aprendê-los. A robótica tem um bom exemplo para esse tipo de treinamento: robôs inteligentes que limpam a casa. Suponha que todo dia o robô encontre poeira de baixo da cama. Este tipo de dado vai fazer com que o robô entenda que todo dia vai ter sujeira de baixo da cama e por isso ele deve, todo dia, limpar a sujeira de baixo da cama.

O semi-supervisionado utiliza parte dos dados rotulados, e como exemplo para este tipo de treinamento, temos as redes sociais que conseguem reconhecer usuários em fotos postadas na rede. Isso é possível porque quando um usuário Y posta uma foto com o usuário X, o algoritmo reconhece que a mesma pessoa está presente em algumas imagens, mas ainda não reconhece como usuário X. Então o usuário Y identifica o usuário X em algumas fotos e a partir deste ponto o algoritmo é capaz de reconhecer o usuário X.

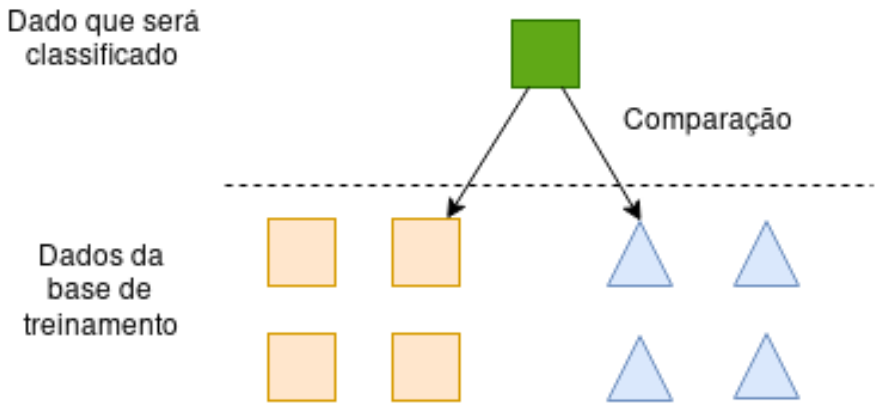
Por último, temos o aprendizado por reforço que utiliza um sistema para que a máquina aprenda sozinha, mas com um fator de recompensa para cada tarefa executada. Assim, a máquina é capaz de aprender o que fazer em cada situação, ou seja, a máquina como um papel de agente realiza tarefas em um ambiente e aprende de acordo com sua política de recompensa.

O aprendizado por lote utiliza dados disponíveis para o treinamento, e este é feito de forma offline e apenas entra em produção depois do treinamento estar completo. Se houver necessidade de atualizar os dados do sistema, o treinamento deve ser refeito do zero com os novos dados. Este tipo de aprendizado consome muito recurso computacional e também muito tempo para executar o treinamento devido aos lotes possuírem uma grande quantidade de dados, mas é recomendado para sistemas nos quais não há um fluxo contínuo ou o treinamento será executado uma vez só. Aqui se encaixa como exemplo o caso da detecção de tumores cerebrais em imagens.

No caso do aprendizado online, o sistema possui aprendizado incremental, ou seja, é possível adicionar novos dados para o treinamento enquanto o sistema opera criando assim um aprendizado mais rápido e menos custoso computacionalmente. Esse tipo de aprendizado é recomendado para cenários que possuem um fluxo contínuo como o caso do filtro de *spam*, pois enquanto o sistema opera, se algum usuário receber um *spam*, basta sinalizar o *e-mail* e o mesmo será reconhecido pelo sistema e incrementado nos dados disponíveis.

Por fim, temos o tipo de funcionamento dos algoritmos. A ideia nesta etapa é generalizar as métricas para poder aplicar em novos dados. Primeiramente, abordando o método de comparação de dados conhecidos com dados desconhecidos, o sistema é capaz de medir a similaridade dos dados conhecidos com o dado desconhecido que deve ser classificado e desta forma o sistema generaliza para cada dado que aparecer para ser classificado.

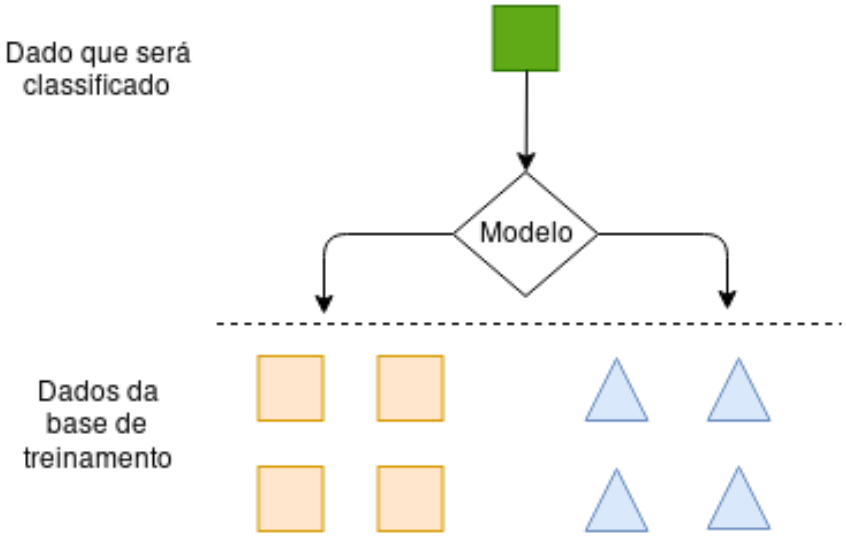
Figura 5 – Generalização por comparação



Fonte: Os autores

Outra maneira de se generalizar a partir dos dados disponíveis é criar um modelo baseado nos dados utilizados para treinamento e utilizar este modelo para classificar novos dados. Por exemplo, se o objetivo de um trabalho é reconhecer gatos e cachorros, deve-se utilizar todo *dataset* de imagens separando gatos e cachorros. Com isso, durante o treinamento é criado um modelo para reconhecer ambos.

Figura 6 – Generalização por modelo



Fonte: Os autores

3.1.2 *Overfitting* e *Underfitting*

Pode haver problemas com os resultados do sistema caso a base utilizada para o treinamento seja pequena, tenha muito ruído ou até mesmo tenha sido construída de maneira incorreta. Além disso, por mais que a base esteja boa, também podem ocorrer erros durante o treinamento que alteram o resultado de forma significativa, como o *overfitting* e *underfitting*.

O *overfitting* acontece quando há muito ruído na base de treinamento, e estes ruídos podem ocorrer por: não realização de um pré-processamento ou não tratamento de dados antes do treinamento; o ruído estar sendo introduzido na amostragem porque a base utilizada para o treinamento é tão pequena que o aprendizado não consegue generalizar os padrões. Para tratar o *overfitting* deve-se considerar como solução: aumentar os dados da base de treinamento, tratar os dados retirando ruído dos mesmos e simplificar o modelo reduzindo o número de atributos ou parâmetros.

O *underfitting* é exatamente o oposto do *overfitting*. Geralmente acontece quando o modelo de aprendizagem é muito simples e não consegue capturar os padrões necessários para classificação da base de treinamento, não sendo possível também generalizar corretamente. Para tratar o *underfitting*, deve-se considerar um modelo com mais parâmetros e características dos dados que não foram consideradas anteriormente, e reduzir as restrições do modelo.

3.2 REDES NEURAIS

Agora que vimos os conceitos de *Machine Learning* podemos entender melhor o termo *Deep Learning*, que de maneira geral é uma subcategoria do *Machine Learning* e trata o desenvolvimento de um sistema de Inteligência artificial utilizando redes neurais, possibilitando solucionar problemas e criar soluções inovadoras como reconhecimento de padrões em imagens, reconhecimento de fala e até auxiliar o ser humano na tomada de decisões em um problema com uma grande quantidade de dados.

As redes neurais podem aprender e modelar relações entre entradas e saídas de dados, isso é feito realizando generalizações dos padrões identificados de uma base de dados para poder realizar predições dos dados de entrada. As redes neurais são sistemas de computação com nós interconectados, chamados de *Perceptron*, que funcionam como os neurônios do cérebro humano. Usando algoritmos, elas podem reconhecer padrões em uma base de dados agrupá-los, classificá-los, e até melhorar os resultados continuamente aprendendo com novos dados que são inseridos no sistema.

Há alguns tipos de redes neurais como:

a) Rede neural artificial

b) Rede neural convolucional

c) Rede neural recorrente

Apesar de citadas as redes existentes, este projeto foi desenvolvido focado na rede neural convolucional, pois esta rede utiliza uma arquitetura bem desenvolvida e adaptada para classificar imagens, o uso dessa rede para classificação de imagem tem um melhor desempenho para treinamento se comparado com a rede neural artificial, pois a tratativa da rede neural multicamada, que são criadas a partir da imagem, é feita de forma diferente.

3.2.1 Redes Neurais Artificiais

Rede Neural Artificial ou RNA, apresenta um modelo matemático inspirado na estrutura neural de animais e seres humanos. Elas adquirem conhecimento de acordo com seu tipo de aprendizado e auxiliam na tomada de decisão.

A tomada de decisão ocorre nas unidades de processamento que compõem a rede neural, essas unidades são conectadas e estão associadas a um determinado peso criando assim algo similar as portas lógicas como *AND*, *OR* e *NOT*.

O comportamento inteligente da rede neural ocorre por causa da interação que as unidades de processamento têm entre si a respeito dos dados que são recebidos como entrada, podendo assim obter um resultado diferente como saída para cada dado de entrada.

A operação de uma unidade de processamento possui o seguinte fluxo:

a) Sinais são apresentados à entrada;
b) Cada sinal é multiplicado por um número, ou peso;
c) É feita a soma ponderada dos sinais que produz um nível de atividade;
d) Se este nível de atividade exceder um certo limite (*threshold*) a unidade produz uma determinada resposta de saída.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos.

Para entender melhor a rede neural devemos aprofundar o conhecimento no funcionamento da unidade de processamento da rede, o *Perceptron*.

3.2.2 *Perceptron*

O *Perceptron* possui uma arquitetura simples, é um modelo matemático que recebe entradas, pode que são multiplicadas por pesos, responsáveis por quantificar a importância de cada entrada, produzindo assim uma saída binária determinada pela soma ponderada dos pesos e das entradas. Veja na figura 7 um exemplo do *Perceptron* com pesos, onde foi considerado os pesos iguais para as entradas por isso temos como saída a função, como também mostrado em (DATA SCIENCE ACADEMY, s.d.):

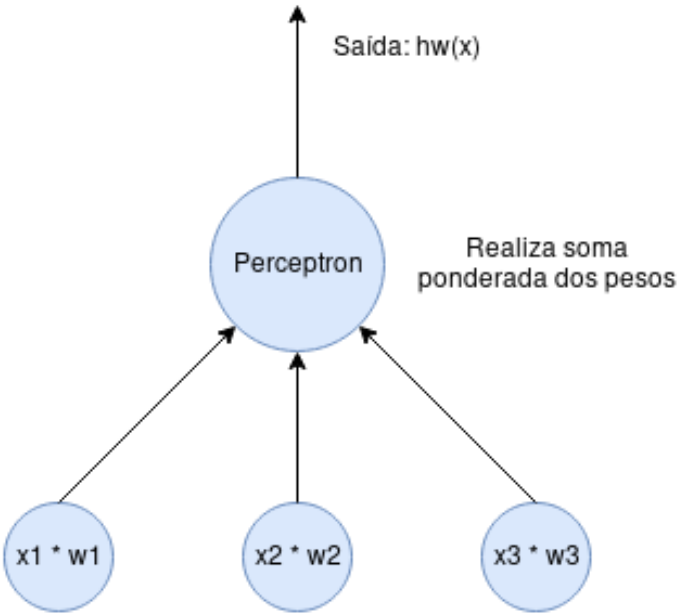
$$h_w(x) = step(w^T * x) \tag{1}$$

Porém a formula para soma ponderada é:

$$z = w_1 * 1 + w_2 * 2 + ... + w_n * n \tag{2}$$

Veja na figura 7 o *Perceptron* recebendo a entrada com os pesos:

Figura 7 – Funcionamento do *Perceptron*



Fonte: Os autores

A soma ponderada realizada no *Perceptron* resulta em um número real menor ou maior que o valor chamado de *threshold*. Veja na figura 8 a representação matemática:

Figura 8 – Representação do *threshold*

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Fonte: (DATA SCIENCE ACADEMY, s.d.)

Apesar de ser possível realizar o cálculo utilizando o *threshold*, há uma maneira de simplifica-lo, substituindo o valor do *threshold* por um valor chamado bias. O bias é equivalente ao valor do *threshold* negativo e considerado uma medida de facilidade para o *Perceptron* produzir uma saída binária. Veja a figura 9 a regra reescrita com o bias, sendo "w" o peso da entrada e "x" o valor da entrada:

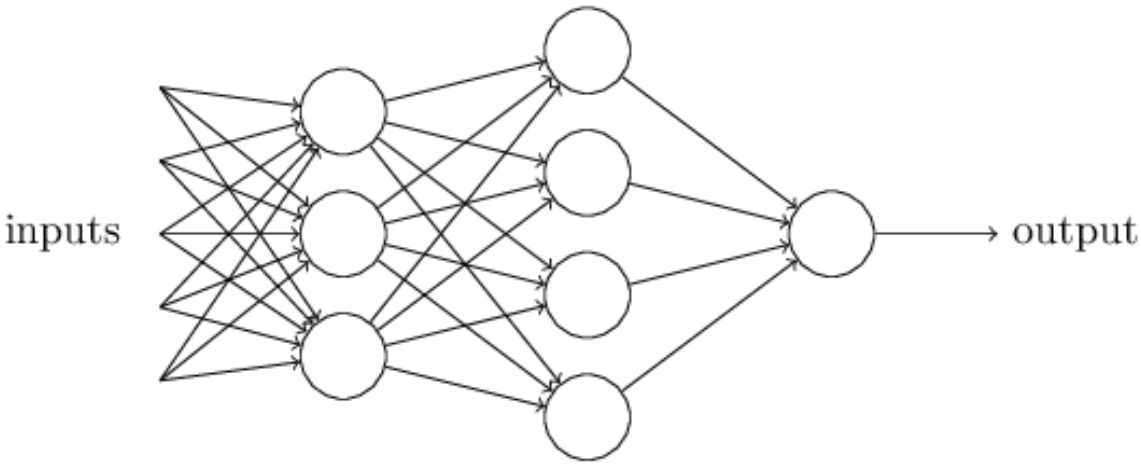
Figura 9 – Representação do bias

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

Fonte: (DATA SCIENCE ACADEMY, s.d.)

Perceptron é uma rede neural de camada única e pode tomar decisões baseado nos dados alimentados na entrada, porém o valor de saída do *Perceptron* pode ser tratado por uma nova camada, ponderando os resultados da camada anterior e criando a possibilidade de tomar decisões mais complexas. Quando utilizamos um *Perceptron* de várias camadas, criamos uma rede de *Perceptron* que por sua vez são chamadas de Rede Neural Artificial ou MLP (*Multi-Layer Perceptron*).

Figura 10 – Rede Neural Artificial ou MLP



Fonte: (DATA SCIENCE ACADEMY, s.d.)

Outra característica importante do *Perceptron*, como mostram as fontes (GÉRON, 2017) e (DATA SCIENCE ACADEMY, s.d.), é a representação de condicionais lógicos e operadores booleanos. Operações lógicas como *AND*, *OR* e *NOT* que necessitam apenas de uma camada de*Perceptron* por se tratar de problemas lineares.

Entretanto nem todos os operadores lógicos são linearmente separáveis, como por exemplo o *XOR*, porém a solução para o problema do *XOR* pode ser implementada com MLP.

Este caso dos operadores lógicos ilustra a necessidade de se utilizar a MLP, porém a MLP também possui suas limitações quando se trata de processamento de imagens, por exemplo a MLP não é capaz de identificar dependências espaciais. Portanto para tratar problemas mais complexos, como processamento de linguagem natural e visão computacional, são utilizadas arquiteturas de rede neural específicas.

Entre as redes neurais existentes, a mais utilizada para tratar problemas relacionados a visão computacional é a rede neural convolucional que ao da MLP, é capaz de identificar dependências espaciais e utiliza técnicas para melhorar a aprendizagem da rede como *backpropagation* e função de ativação.

3.2.3 *Backpropagation*

Backpropagation é um algoritmo utilizado para melhorar o aprendizado da rede alterando o peso e o bias para classificar corretamente uma saída. O *Perceptron* segue o modelo de propagação direta, que consiste em processar as entradas do neurônio resultando em uma saída e esta ser processada em outros neurônios da próxima camada até o sistema atingir a saída final, porém o resultado pode não ser o resultado previsto então o erro é retro-propagado da camada de saída até a camada de entrada e os pesos e o bias vão sendo modificados, mas alterar o peso e o bias desde a primeira camada pode fazer a rede classificar corretamente alguma saída que estava errada assim como pode fazer também a rede classificar erroneamente alguma saída que estava correta, portanto para evitar este problema existe a função de ativação.

3.2.4 Função de ativação

A função de ativação é a transformação não linear dos dados de entrada, onde a saída de cada neurônio é transformada por essa função e enviada para próxima camada de neurônios. A função de ativação torna possível o aprendizado de tarefas complexas que transformações lineares não são capazes de executar, como classificar imagens e texto. Há alguns tipos de funções de ativação, as mais populares são:

a) Função de Etapa Binária
b) Função Linear
c) Sigmóide
d) Tanh

e) ReLU

f) Leaky ReLU

g) Softmax

Neste projeto foi utilizada a função ReLU (*Rectified Linear Unit*), esta função é utilizada apenas nas camadas ocultas e converte a entrada negativa de cada neurônio para zero, não ativando este neurônio uma vez que a função é definida como:

$$f(x) = max(0,x) \tag{3}$$

Além de ser a função mais utilizada para maioria dos casos para modelar uma rede neural, também traz como vantagem sobre as outras funções a não ativação dos neurônios ao mesmo tempo, isso torna o algoritmo menos custoso computacionalmente.

3.2.5 Redes Neurais Convolucionais

A Rede Neural Convolucional é chamada também de CNN (*Convolutional Neural Network*) ou ConvNet, é um algoritmo de *Deep Learning* que pode captar uma imagem de entrada e atribuir pesos, que podem ser aprendidos, a vários aspectos da imagem e ser capaz de diferenciá-los.

Esta arquitetura também é capaz de identificar dependências espaciais em uma imagem utilizando filtros que também são chamados de mapas de recurso. As redes neurais convolucionais usam três conceitos básicos:

a) Campos receptivos locais;

b) Pesos compartilhados;

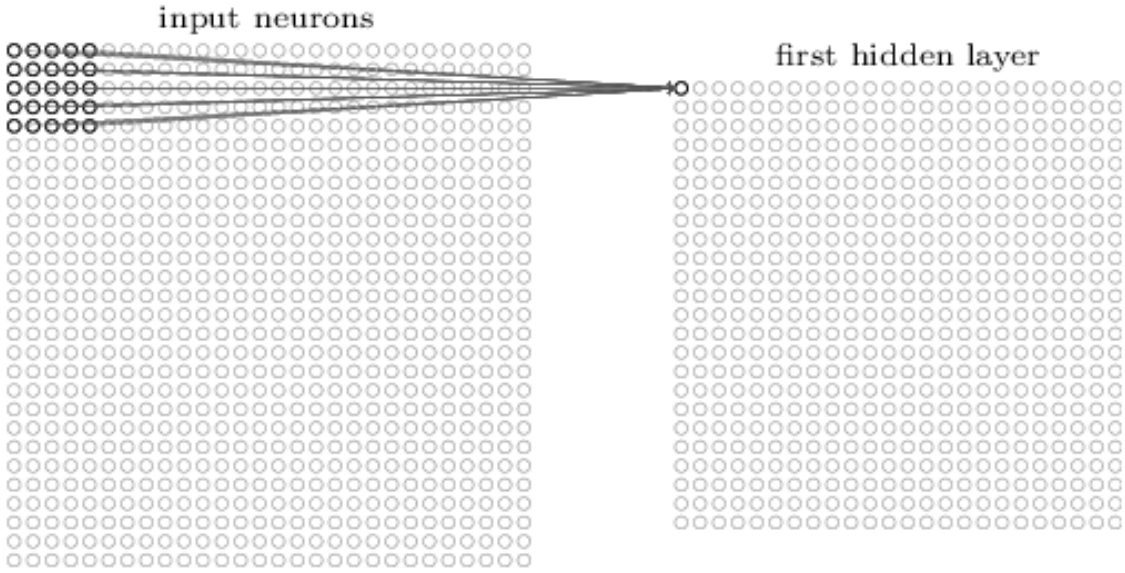
c) *Pooling*.

Os campos receptivos locais são uma janela nos *pixels* de entrada onde cada conexão aprende um peso e o neurônio oculto aprende um bias geral. Como exemplificado em (DATA SCIENCE ACADEMY, s.d.), para ilustrar o campo receptivo local a figura 11 mostra os neurônios de entrada e cada um deles possui o valor da intensidade do pixel de uma imagem, ou seja, se uma imagem possui 28 × 28 *pixels*, significa que a rede tem 784 neurônios de entrada.

Esses neurônios de entrada serão conectados aos neurônios da camada oculta que representarão uma região da camada de entrada, por exemplo, uma região de 25 *pixels* é representada

por 5 x 5 neurônios da camada de entrada, essa é a região chamada de campo receptivo local. Veja na figura 11 a ilustração da construção da camada oculta:

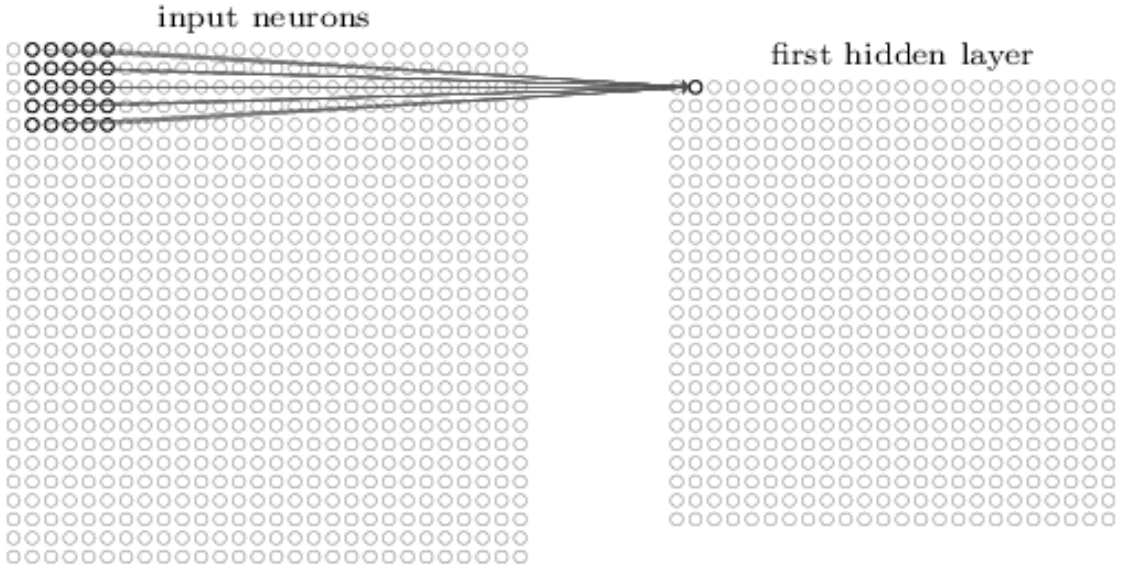
Figura 11 – Primeiro passo para construção da camada oculta



Fonte: (DATA SCIENCE ACADEMY, s.d.)

Considerando a imagem 11 a região destacada se move 1 pixel para direita e assim a camada oculta vai se construindo, chamada de camada convolucional. Veja na figura 12:

Figura 12 – Segundo passo para construção da camada oculta

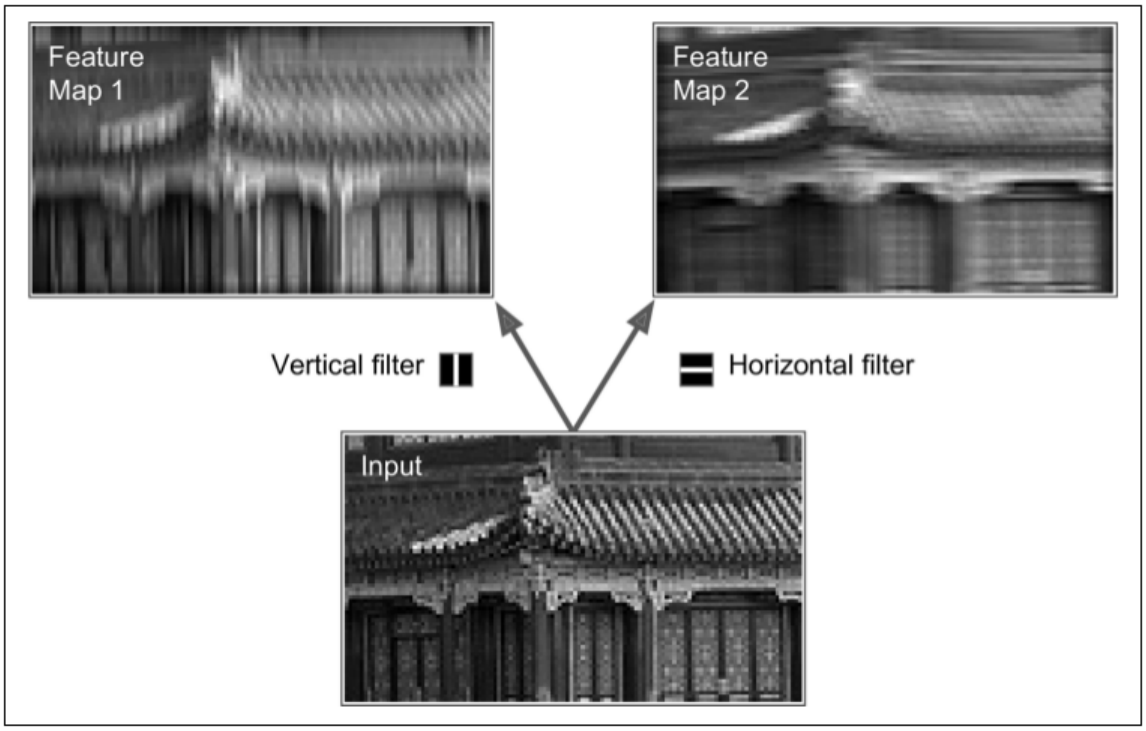


Fonte: (DATA SCIENCE ACADEMY, s.d.)

Não é uma regra caminhar *pixel* a *pixel* da imagem, como visto no exemplo, esse passo de *pixels* pode ser controlado através de um parâmetro chamado *stride length*.

Sobre os pesos compartilhados, esses costumam definir um *kernel* ou filtro, em outras palavras os pesos definem um mapa de recursos da imagem. Mapas de recursos destacam áreas da imagem que são mais similares ao filtro que os neurônios da imagem estão utilizando, durante o treinamento a rede convolucional aprende a combiná-los de forma única. A figura 13 ilustra a aplicação dos filtros horizontal e vertical:

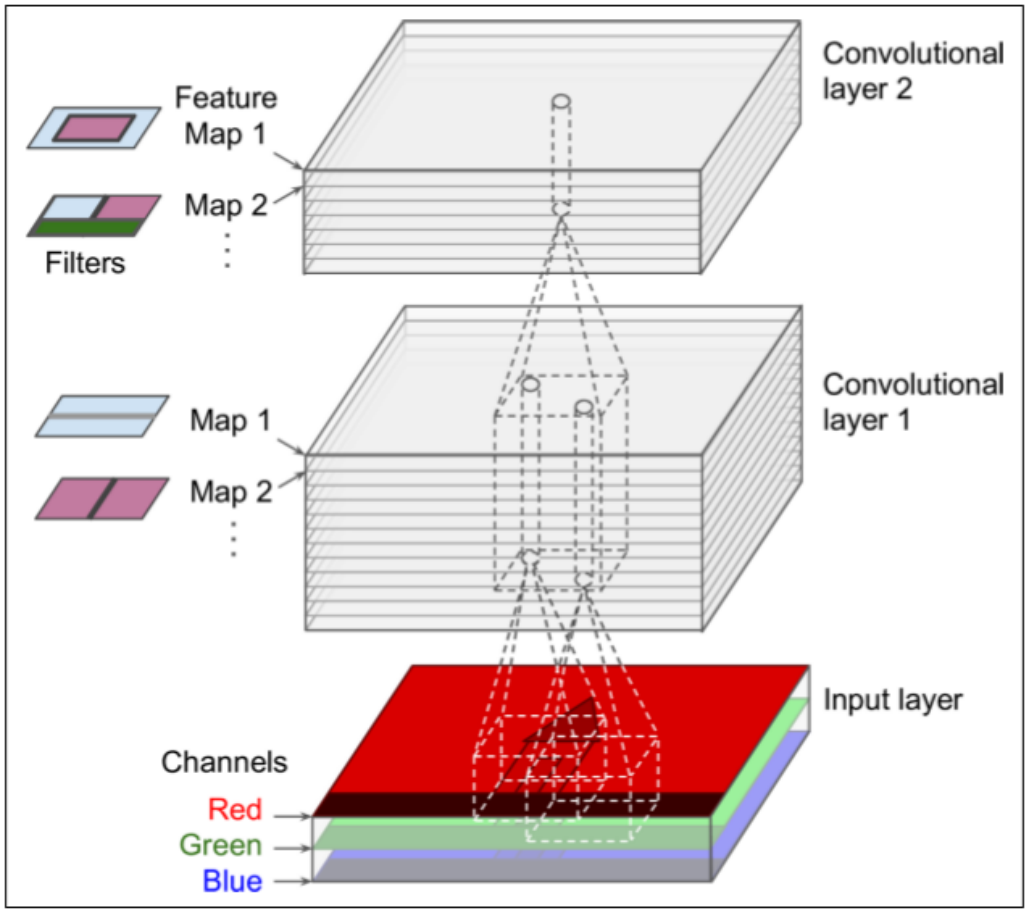
Figura 13 – Filtros horizontal e vertical de uma imagem



Fonte: (GÉRON, 2017)

Além dos filtros vertical e horizontal há outros que possuem seus próprios mapas de recurso em suas camadas convolucionais, como a escala de cores RGB por exemplo.

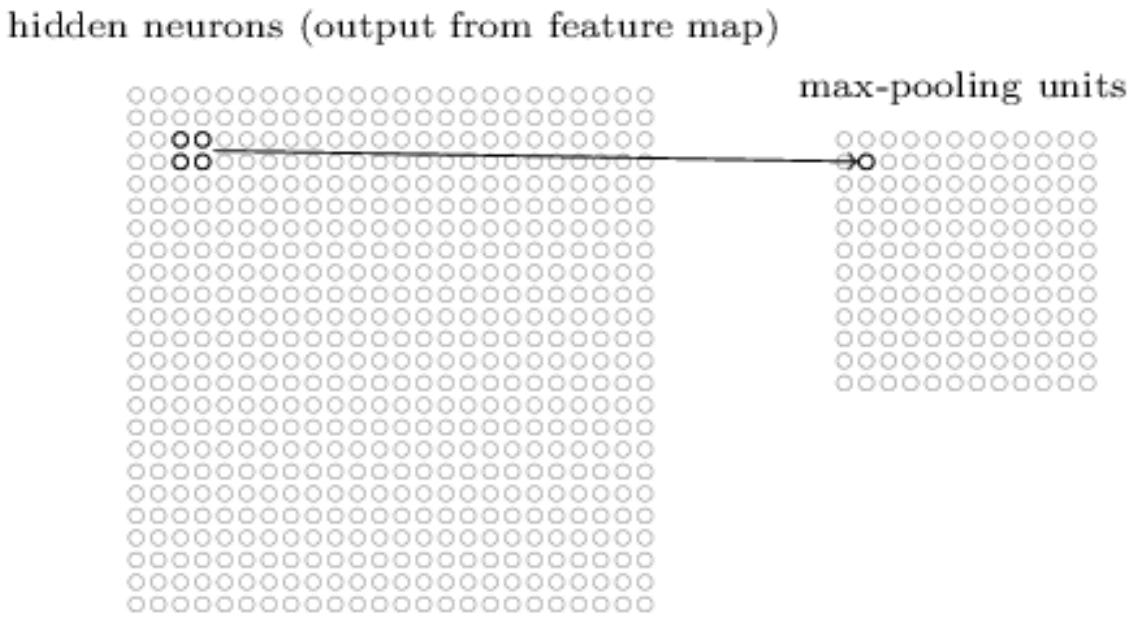
Figura 14 – Mapas de recurso RGB



Fonte: (GÉRON, 2017)

A camada de *pooling* recebe cada saída do mapa de recursos da camada convolucional e prepara um mapa de recursos condensado. Cada unidade na camada de *pooling* é conectada a uma região da camada de um mapa de recursos, ou seja, para cada mapa há uma camada de *pooling*.

Figura 15 – Construção da camada de *pooling*



Fonte: (DATA SCIENCE ACADEMY, s.d.)

3.3 *HAAR-CASCADE*

O *Haar-cascade* É um algoritmo classificador capaz de identificar objetos após algumas etapas de treinamento de identificação. Pode-se adaptar o *haar-cascade* para detectar rostos, características faciais e corporais. É utilizado junto bibliotecas públicas como o *OpenCV*.

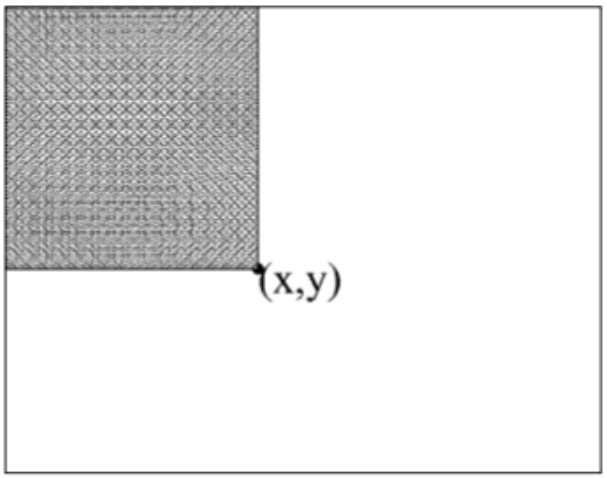
3.3.1 Algoritmo de Viola e Jones

Viola e Jones (2004) trouxeram para a academia uma nova proposta para detecção de objetos em imagens. A motivação deles era realizar um reconhecimento facial de forma rápida e eficiente. Há 3 conceitos principais no trabalho de Viola e Jones.

O primeiro conceito é a integral da imagem. Neste conceito, o sistema de detecção não trabalha diretamente com intensidades de imagem, mas sim trata a soma de áreas da imagem em uma tabela muito semelhante ao trabalho divulgado por *Frank Crow* em 1984 sobre o mapeamento de texturas, que avalia os valores dos *pixels* de uma sub-região retangular de uma imagem em escala de cinza.

Com a integral da imagem é possível subtrair os valores dos *pixels* de uma sub-região retangular com outra, obtendo a diferença de intensidade luminosa entre essas sub-regiões. Este método é chamado de característica tipo *Haar* e com isso é possível identificar padrões. Essa diferença de intensidade luminosa pode ser aplicada na face, para o reconhecimento de características faciais, pois segundo Viola e Jones, a região dos olhos, por exemplo, é mais escura que a região das bochechas. Com um algoritmo que reconheça esses padrões é possível reconhecer características da face. Veja na imagem 16, calcula-se a integral da imagem da área delimitada, esta sub-região se movimenta da esquerda para direita e de cima para baixo:

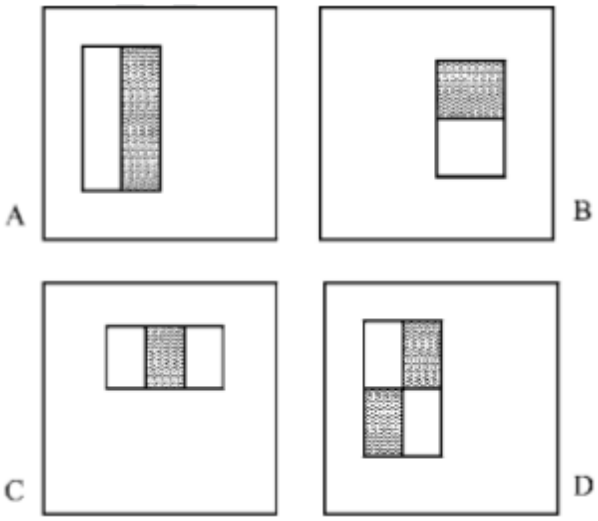
Figura 16 – Funcionamento da varredura das sub-regiões na imagem



Fonte: (VIOLA; JONES, 2004)

Veja na figura 17, para cada "letra"da figura calcula-se a integral da imagem, isso é feito a partir dos pontos que formam cada retângulo, por exemplo, a letra C possui 8 pontos. As características tipo *Haar* são calculadas subtraindo-se a soma dos valores dos *pixels* da região branca da soma dos valores dos *pixels* da região preta.

Figura 17 – Representação das características tipo *Haar*



Fonte: (VIOLA; JONES, 2004)

O segundo conceito do algoritmo de Viola e Jones é a construção de um classificador simples e eficiente, para garantir uma classificação rápida no processo de aprendizagem. Deve-se excluir grande maioria das características disponíveis e concentrar-se em um pequeno conjunto de características críticas, isso é possível ser realizado utilizando um algoritmo de aprendizagem chamado *AdaBoost* que tem como função construir um classificador forte combinando o resultado de classificadores fracos. No *AdaBoost* cada classificador fraco é um estágio do processo que seleciona o próximo classificador fraco.

Na figura 18 há 2 características selecionadas pelo *AdaBoost*, a primeira mede a diferença de intensidade entre os olhos e as bochechas e a segunda mede a intensidade entre os olhos e o nariz.

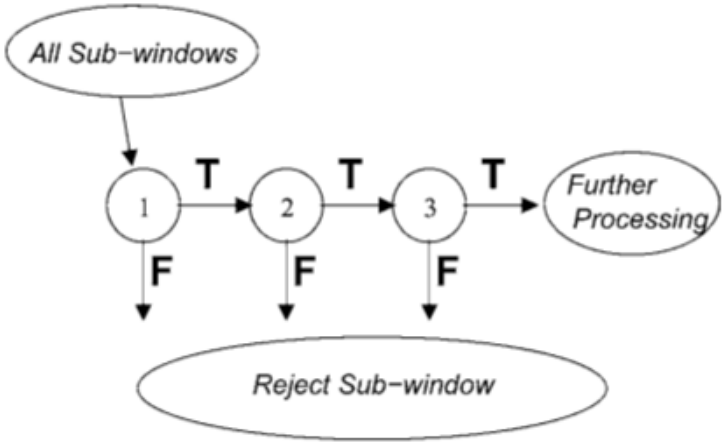
Figura 18 – Identificação das características com o *Haar*

Fonte: (VIOLA; JONES, 2004)

O terceiro conceito consiste na combinação dos classificadores sucessivos em uma estrutura de cascata, concentrando assim a atenção em regiões da imagem que contém as características procuradas. A estrutura de cascata trabalha com estágios em cada região da imagem, entretanto se as características procuradas não forem encontradas em um determinado estágio, esta região é descartada para os próximos estágios, isso aumenta drasticamente a velocidade do processamento da imagem.

A figura 19 exemplifica o funcionamento dos classificadores em cascata, sendo que a imagem é a entrada no primeiro estágio, o "T"o resultado positivo e encaminhamento para o próximo estágio e o "F"quando o resultado é negativo em um estágio e a imagem é rejeitada.

Figura 19 – Classificador em cascata



Fonte: (VIOLA; JONES, 2004)

O algoritmo de Viola-Jones foi desenvolvido com o objetivo de detectar faces e o mesmo apresenta um baixo número de falsos positivos, porém apenas quando as imagens possuem faces de forma frontal, pois o algoritmo falha na detecção de faces de perfil ou que possuem objetos e outras características como óculos de sol e barbas muito grandes, portanto para este projeto este é um ponto que deve ser tratado para aumentar a precisão do algoritmo.

3.3.2 Detecção de objetos com *haar-cascade*

Na visão computacional quando uma imagem é processada, tem como objetivo realçar sua qualidade. Nesse processo são removidos ruídos e outras imperfeições, então são detectados padrões e formas para poder determinar o que há na figura, que é a detecção de objetos.

Uma forma de identificar objetos é através das cores, entretanto esse método não pode ser aplicado para maior parte das situações do mundo real, por causa das condições de ilumina-ção do ambiente em que a foto foi tirada e também porque se basear no valor RGB do pixel é processo custoso computacionalmente.

Esses problemas foram direcionados para características do tipo *haar* que considera a vizinhança de regiões retangulares, ele soma a intensidade do pixel em cada região e calcula a diferença, subseções, entre as somas. Há várias características do tipo *haar*, a utilizada nesse projeto será a característica para detecção de rostos.

Sabendo as características que procuramos, deve-se criar uma lista de estágios para um aprendizado fraco. Essa lista é chamada de classificadores em cascata ou "*cascade classifiers*", por isso o nome do algoritmo é *haar-cascade*.

Cada estágio de uma região específica é classificado em positivo ou negativo. Se o objeto foi encontrado ou não e depois disso, se for negativo a janela se move para outra região, se o resultado for positivo é feito o próximo estágio de classificação.

3.4 *OPENCV*

O *OpenCV (Open Source Computer Vision Library)*, de acordo com o site oficial (SITE. . ., s.d.), é uma biblioteca multiplataforma para desenvolvimento de aplicações que abordam o tema de visão computacional, pois possui módulos para o processamento de imagens, vídeos, inter-face gráfica do usuário (*GUI*) e controle de periféricos como mouse, teclado e câmera.

O *OpenCV* está sob a licença do *BSD* e é uma biblioteca livre tanto para o uso acadêmico quanto para o uso comercial, foi desenvolvido para obter eficiência computacional com foco em aplicações de tempo real e também possui suporte para as linguagens *C++*, *Python* e *Java*.

Essa biblioteca, segundo sua documentação (SITE..., s.d.), possui uma estrutura modular, o que facilita o uso de cada módulo do *OpenCV*, pois basta incluir o modulo no cabeçalho do código e respeitar a sintaxe da linguagem utilizada. Os módulos também são utilizados para detecção de objetos e faces, alguns deles que serão utilizados neste projeto são:

a) Core: define estrutura de dados como vetores multidimensionais, chamados de "Mat"e funções que serão utilizadas por outros módulos;
b) *Imgproc*: processa imagens incluindo filtros, transformação geométrica da imagem, conversão de cor no espaço e etc.;
c) *Objdetect*: detecção de objetos pré-definidos como faces, olhos, pessoas, carros e etc.

3.5 *KERAS*

Uma API de redes neurais de alto nível construída em Python capaz de funcionar em cima do *TensorFlow*, *CNTK* ou *Theano*, o desenvolvimento do *Keras* tem como foco permitir uma utilização mais prática e rápida, sendo possível ir da ideia ao resultado. A utilização da biblioteca do *Keras* permite, que seja possível criar protótipos rápido e fáceis, pois contém uma facilidade no uso da modularidade extensibilidade, contém o suporte para redes convolucionais e redes recorrentes e a combinação entre elas, é compatível com o *Python 2.7 - 3.6* e um funcionamento perfeito com CPU (Unidade Central de Processamento) e GPU (Unidade de Processamento Gráfico) de acordo com (CHOLLET et al., 2015).

O *Keras* consiste em oferecer uma facilidade e agilidade, que minimiza o número de ações do usuário para casos de uso e fornece um *feedback* claro sobre o erro do usuário, o *Keras* também utiliza a modularidade tendo módulos independentes e totalmente configuráveis, sendo possível combinar e criar novos modelos, alguns módulos existentes:

a) Camadas neurais;
b) Funções de ativação;
c) Otimizadores;
d) Esquemas de inicialização.

Usando o conjunto de módulos, criação de novos módulos a partir da combinação e facilidade para adicionar novos módulos, o *Keras* permite a flexibilidade e expressividade para cada utilização tornando adequada a utilização em pesquisas avançadas. Para as configurações dos modelos não existe um formato descritivo separado segundo o (CHOLLET et al., 2015), o Python é utilizado na escrita dos modelos deixando compacto e fácil para depurar permitindo a facilidade e extensibilidade.

4 METODOLOGIA

Conforme abordado anteriormente, este trabalho trata do problema de detecção de crianças em imagens utilizando um modelo de rede neural convolucional. Esse capítulo apresenta detalhadamente a metodologia proposta.

O modelo geral proposto pode ser visualizado na Figura 20. Note que esta figura é dividida em dois domínios: Faces e Cenas. Uma das contribuições desse trabalho é a utilização de uma base supervisionada de faces, geradas a partir de imagens 3x4, para o treinamento de uma rede neural convolucional. Essa mesma rede é então utilizada para detecção de crianças em quaisquer tipos de cenas, não somente 3x4.

4.1 BASE DE DADOS

Essa seção descreve as bases de dados utilizadas para o treinamento e validação dos resultados.

4.1.1 *Dataset 01*

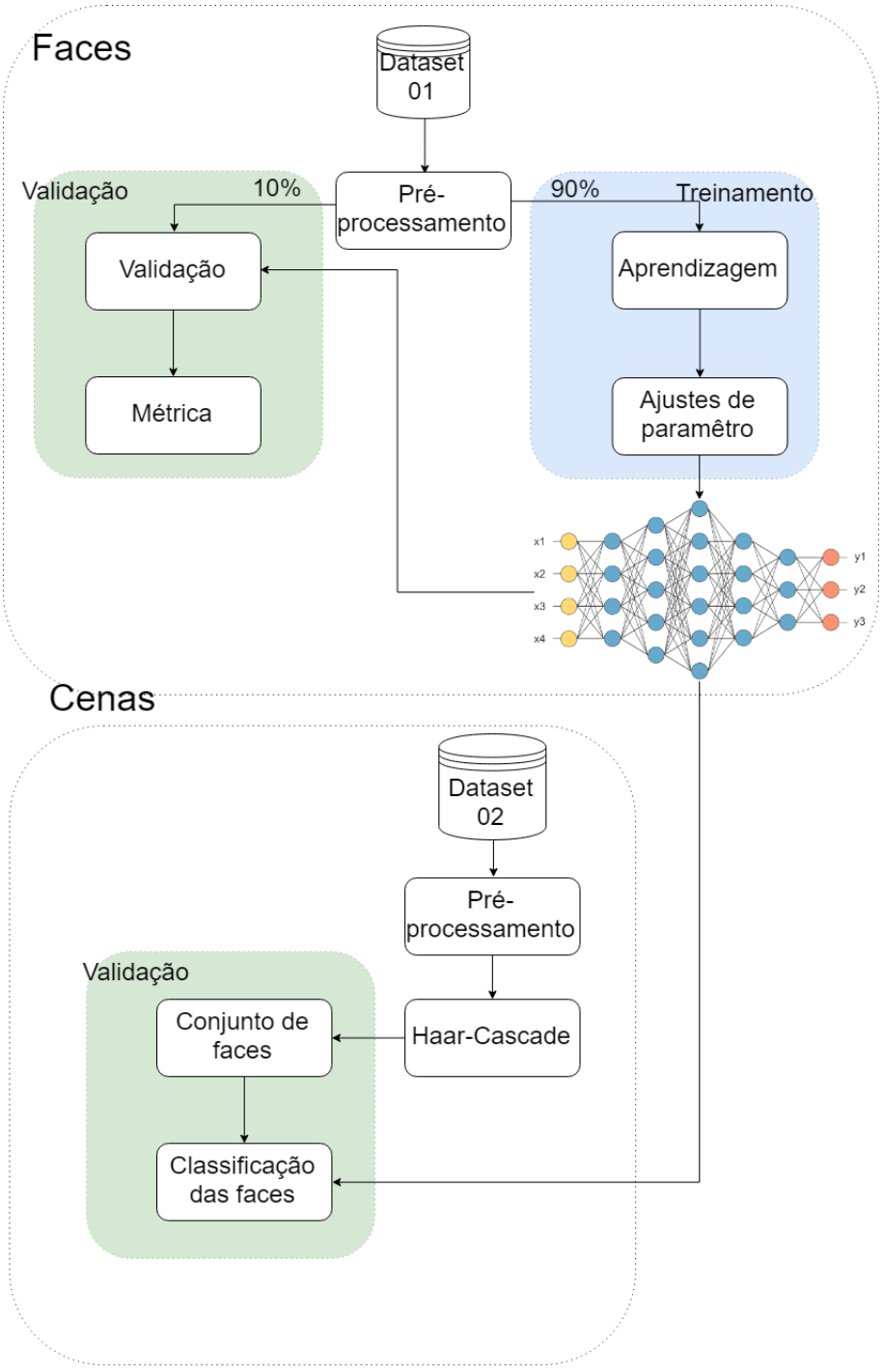
O primeiro *dataset*, passo número 1 apresentado na Figura 20, contém as imagens utilizadas para realizar o treinamento da rede convolucional, são imagens 3x4 com pouca angulação. Esse *dataset* foi construído a partir de duas bases (BIANCO, 2017) e (KINFACEW, s.d.), a primeira fonte de imagens contém imagens de pessoas famosas em duas épocas diferentes, de quando eram crianças e adultas.

A segunda fonte contém imagens com uma relação de parentesco entre as crianças e os adultos presentes nas imagens, essas relações são de mãe e filho, mãe e filha, pai e filho, pai e filha. Tendo assim um total de 4.755 imagens para realizar o treinamento da rede convolucional, sendo 2.587 de faces de adultos e 2.168 de faces de menores de idade.

4.1.2 *Dataset 02*

O segundo *dataset* é composto por imagens diversas podendo ou não conter pessoas, esse *dataset* é utilizado para identificar a *accuracy* do *haar-cascade*, já que é o *haar-cascade* que irá identificar as faces para enviar para a rede neural convolucional. Esse *dataset* foi extraído *Kaggle* contendo 6.899 imagens.

Figura 20 – Fluxograma do projeto



Fonte: Os autores

4.2 FACES

A etapa de faces apresentada no fluxograma da imagem 20, é onde ocorro todo o treinamento e métrica da rede neural convolucional, utilizando o *dataset 01* que contém apenas as faces de pessoas.

4.2.1 Treinamento

Para o treinamento da rede convolucional foi necessário realizar um pré-processamento na base de imagens, com o objetivo de melhorar o processo de treinamento. Primeiramente, todas as imagens do *dataset* 01 foram redimensionadas para uma escala de 150x150 *pixels* e convertidas em uma escala de cinza. Esta etapa é necessária pois a rede convolucional somente aceita entradas de tamanho fixo, neste caso, 22.500 $(150 * 150)$ entradas. Além disso, a conversão para escalas de cinza foi feita para diminuir o número de entradas. Dessa forma, um único canal de luminância (escala de cinza) é fornecido para a rede.

Com o pré-processamento realizado, a rede convolucional pode realizar a etapa de treinamento. Dessa forma, 90% da base de imagens é utilizada nesta etapa e, a partir dos dados supervisionados, dois vetores são criados: um contendo as imagens; e outro contendo os dados das supervisões. Esses dados supervisionados indicam para cada imagem se há ou não a presença de crianças.

A partir desse subconjunto da base, os dados de entrada e saída esperada são enviados para a rede convolucional. Aqui, foram testados diversos meta-parâmetros da rede para obter melhores resultados de classificação.

Esses parâmetros são definidos nas funções da etapa de aprendizagem, como número de neurônios, função de ativação, dimensão do kernel, entre outros. Os parâmetros utilizados nesse rede neural foram escolhidos empiricamente. Para essa rede convolucional, foi decidido utilizar a função de ativação *ReLU*, por ser uma das funções mais utilizadas para modelos não-lineares.

A utilização da função *Dropout* foi utilizado com o objetivo de diminuir o *overfiting*, pois a função *Dropout* inibi uma porcentagem de neurônios parametrizada, com isso o número de neurônios diminui tendo assim praticamente uma nova rede neural apresentando um novo resultado, com isso é feito a média desses resultados trazendo assim possivelmente uma diminuição do *overfiting*.

As janelas de convolução foram muito importantes para o treinamento, para que seja possível retirar as melhores características das imagens. A dimensão da janela de convolução é um dos principais parâmetros que interferem na qualidade da classificação. Para essa rede, foi utilizada uma janela de convolução um pouco maior do que a convencionais para poder extrair características maiores das crianças, tais como nariz e boca, e depois retirar os detalhes dessas janelas maiores, assim foi iniciado com uma janela de 7x7 *pixels*.

4.2.2 Validação

Para realizar a validação final, passos 6 e 7 da imagem 20, foi utilizado o método de validação cruzada, *cross-validation*.separando a base em subconjuntos e rearranjando os subconjuntos tanto para treinamento quando para validação, para conseguir resultados consistentes e não enviesados. Ao final dos experimentos, é computada um média aritmética de todos os experimentos.

Para a etapa de métrica, número 7 da imagem 20, no método de predição foi utilizado um peso de 0.5, sendo menor que 0.5 identificado como criança e maior que 0.5 identificado como adulto. Empiricamente foram os que trouxeram melhores resultados.

4.3 APLICAÇÃO EM CENAS

Após a conclusão do treinamento da rede neural e sua validação, é realizado o processo de validação para imagens contendo cenas e não apenas faces de pessoas. Por isso é utilizado o *haar-cascade* que fará a identificação dessas faces nas cenas.

O passo de número 9, como mostra a Figura 20, tem como objetivo realizar o pré-processamento das imagens, neste caso as imagens foram passadas para uma escala de cinza e redimensionadas para 150x150 *pixels*. Com o passo 9 finalizado, é possível realizar o passo 10, *Haar-Cascade*, que tem como objetivo identificar as faces de pessoas em imagens, sendo de crianças ou não, assim separando todas as imagens de faces encontradas.

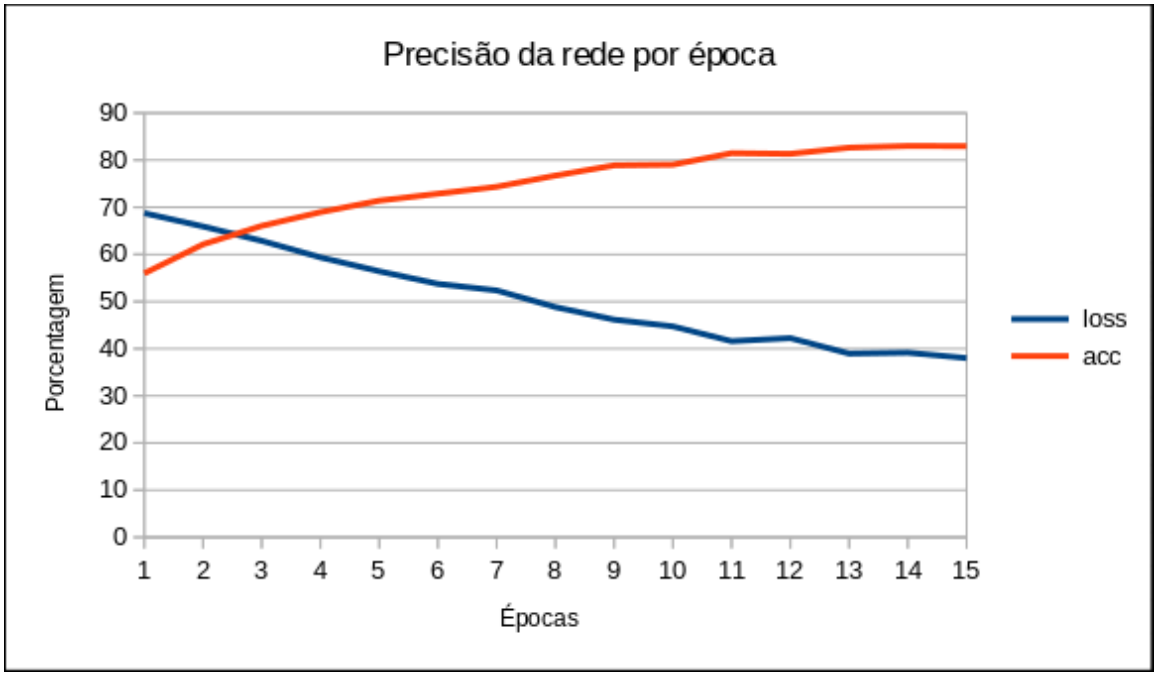
Com as faces identificadas pelo *Haar-Cascade*, passo 11 na figura 20, é possível utilizar a rede neural convolucional para classificar as faces como crianças ou não. Identificando criança, aquela imagem de cena é separada em um diretório onde só contenham cenas com criança e uma outra pasta apenas com as cenas sem crianças.

5 RESULTADOS

Neste capítulo, será apresentado os resultados obtidos a partir dos experimentos realizados, tanto para a rede convolucional quanto para o *haar-cascade*. Os *datasets* utilizados foram o já mencionados anteriormente, *dataset 01* para a rede convolucional e *dataset 02* para o *haar-cascade* da imagem 20.

Com o objetivo de verificar se o treinamento está de fato sendo feito e convergindo, foram extraídas duas métricas de um experimento: a função de custo ($loss$) e a acuracidade (acc). O gráfico da Figura 21 mostra que a função de custo decresce com o passar das épocas e a taxa de acerto aumenta. Isso é um forte indício de que o modelo está aprendendo.

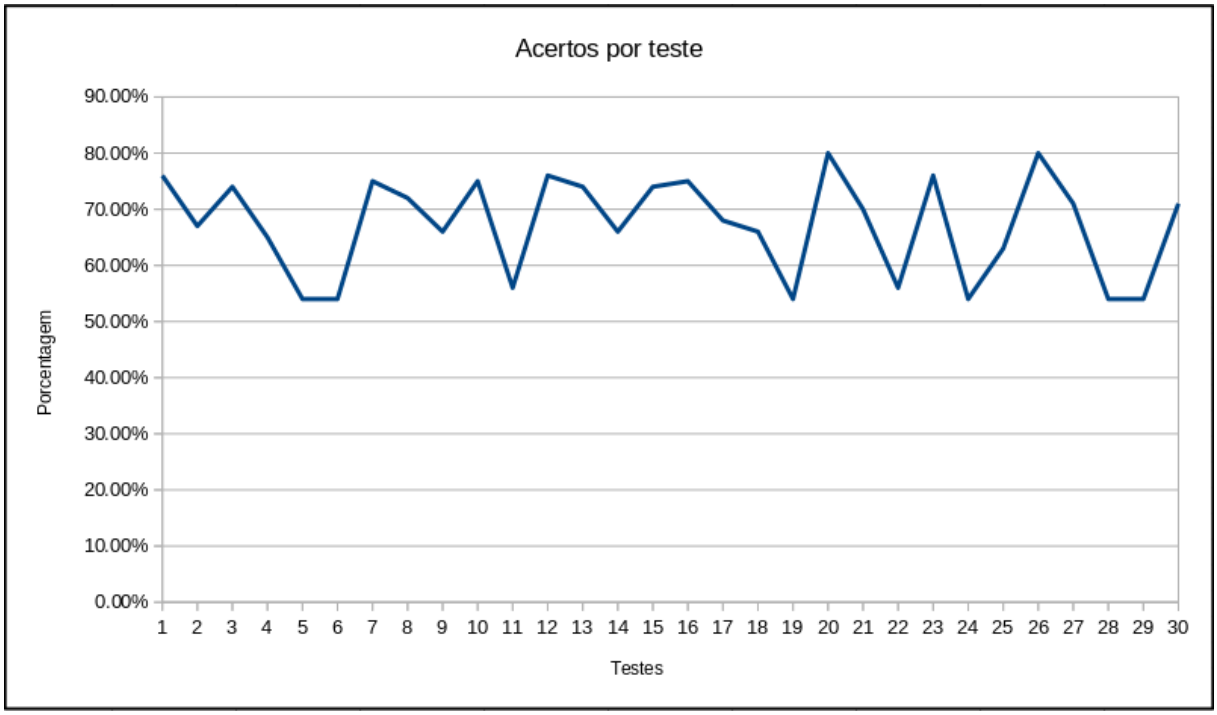
Figura 21 – Evolução do aprendizado da rede por época



Fonte: Os autores

O *cross-validation* foi realizado com 70% da base para treinamento e 30% para validação, e depois a base de treinamento e testes eram rearranjadas. Inicialmente, foram executados 30 testes que gerou o gráfico da Figura 22. Com isso, foi possível extrair a média de acerto de 67% e o desvio padrão de 8%.

Figura 22 – Porcentagem de acerto com 30 testes

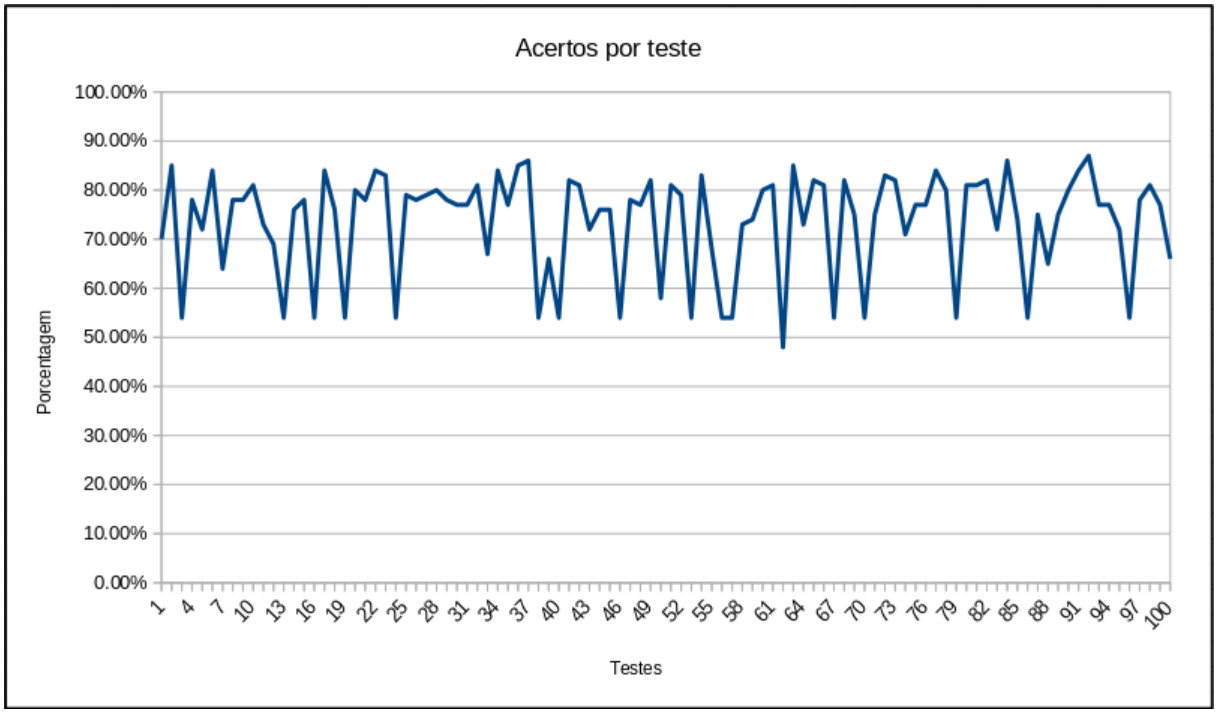


Fonte: Os autores

De acordo com essa imagem, pode-se observar que o desvio-padrão do acerto é alto. A hipótese que é considerada aqui é que as 15 iterações de treinamento da rede neural não foram suficientes para ajustar a rede por completo. De fato, a literatura utiliza valores maiores de iterações, na grandeza de centenas ou milhares. Contudo, a decisão por utilizar 15 iterações veio pelo fato de poupar recursos computacionais, uma vez que a quantidade de variáveis do modelo é muito grande.

Para verificar o quanto o treinamento da rede neural é sensível à quantidade de exemplos (imagens de treinamento), foram feitos 100 novos experimentos mas, dessa vez, utilizando 90% da base para treinamento e 10% para validação. Como mostra o gráfico da Figura 23, a média de acerto foi de 73,55% e o desvio padrão de 10,37%.

Figura 23 – Porcentagem de acerto com 100 testes



Fonte: Os autores

Note que os resultados de ambos os experimentos são similares. Isso é forte indício de que a rede é pouco sensível ao número de entradas. Esse resultado ainda reforça a hipótese do experimento anterior, onde o baixo número de iterações foram insuficientes para o treinamento, mas mesmo assim, geraram alta taxa de acerto.

Para o *haar-cascade*, foram realizados testes empíricos, utilizando o *dataset 02* e três cascatas para detecção facial frontal. Para cada cascata do *haar-cascade* foi obtido uma porcentagem de acerto, para duas das cascatas utilizadas a porcentagem de acerto foi de 97% e para uma delas 98%.

6 CONCLUSÃO

Este trabalho propõem e implementa um modelo computacional misto baseado em dois submodelos: redes neurais convolucionais e *Haar-Cascade*; para detecção de crianças em imagens. Primeiramente, é feita uma detecção facial na imagem com o *Haar-Cascade*. Ao encontrar uma face, a CNN entra em ação para classificar se a face detectada é uma criança ou não.

Antes de juntar os dois algoritmos, foram realizados testes para medir a porcentagem de acerto de ambos os sub-modelos. Para o *Haar-Cascade* foram realizados testes empíricos em três cascatas para detecção facial frontal. Para cada cascata do *Haar-Cascade*, foi obtido uma porcentagem de acerto, sendo que duas das cascatas utilizadas a porcentagem de acerto foi de 97% e para uma delas 98%.

Quanto à CNN, foram realizadas 2 baterias de testes. A primeira consistiu em 30 testes utilizando uma divisão de 70% e 30% da base, respectivamente para treinamento e predição. Nesta etapa se obteve uma média de acerto de 67%.

Para obter resultados mais consistentes e verificar o quanto a rede depende da base de treinamento, foi feita uma nova bateria de testes, porém com 100 testes desta vez. Dessa vez, a divisão da base foi alterada para 90% de treinamento e 10% de validação. Desta vez, se obteve uma média de acerto de 73,55%.

Por falta de uma base supervisionada de cenas com crianças, os sub-modelos foram testados de forma independente. Contudo, a probabilidade conjunta de acerto ainda se mantém elevada, da ordem de 72% ($98\% \times 75\%$).

A partir desses resultados, é possível introduzir o sistema de detecção de criança por imagem em um sistema de *Marketplace*. Utilizando o sistema no momento de cadastro do usuário, forçando o usuário a enviar uma foto tirada na hora do cadastro, assim o sistema faria o reconhecimento daquela foto, permitindo ou não o usuário a finalizar o cadastro.

Como pode-se observar no Capítulo 5, os gráficos mostram que a rede está obtendo resultados consistentes e que, para detectar crianças em imagens através da detecção facial, utilizar a junção do *Haar-Cascade* com a CNN é um caminho viável.

Contudo os resultados deste trabalho podem ser melhorados, pois houveram limitações de hardware que dificultaram o treinamento da CNN. A máquina utilizada para o desenvolvimento desse sistema possui 8 GB de memória RAM e processador com CPU de 2.3 GHz. Por possuir pouca memória RAM e não possuir GPU, a etapa de treinamento com 100 testes demorou cerca de 5 dias, utilizando 15 épocas.

A codificação deste trabalho é divida em três passos: treinamento e testes da CNN de forma automatizada; Testes automatizados do *Haar-Cascade*; Junção do *Haar-Cascade* com a CNN. Os códigos também estão publicados em (MOURA, 2018).

6.1 PASSOS FUTUROS

Como mencionado anteriormente, na literatura o número de iterações do treinamento da rede neural, fica na casa de centenas ou milhares. Por esse motivo essa melhoria na implementação da rede convolucional, melhore os resultados do treinamento, diminuindo o desvio padrão e aumentando a porcentagem de acerto.

Após as melhorias para a detecção das crianças em imagens, um próximo passo é implementar o reconhecimento do ambiente da imagem. Reconhecendo o ambiente, é possível identificar se é um ambiente de risco para uma criança. Com essas informações, detecção de criança e reconhecimento de ambiente, e implementado em uma câmera de vigilância. é possível diminuir os riscos para uma criança, como por exemplo uma construção, linha ferroviária entre outras áreas de risco.

REFERÊNCIAS

ANDA, Felix et al. Evaluating Automated Facial Age Estimation Techniques for Digital Forensics. In: IEEE. 2018 IEEE Security and Privacy Workshops (SPW). [S.l.: s.n.], 2018. p. 129–139.

BIANCO, Simone. Large Age-Gap Face Verification by Feature Injection in Deep Networks. **Pattern Recognition Letters**, v. 90, p. 36–42, 2017.

CHOLLET, François et al. **Keras**. [S.l.: s.n.], 2015. https://keras.io.

DATA SCIENCE ACADEMY. **Site oficial da Deep Learning Book Brasil**. [S.l.: s.n.]. Disponível em: <http://deeplearningbook.com.br>.

GÉRON, Aurélien. **Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems**. [S.l.]: "O'Reilly Media, Inc.", 2017.

KINFACEW. **KinFaceW**. [S.l.: s.n.]. Disponível em: <http://www.kinfacew.com/index.html>.

LI, Haoxiang et al. A convolutional neural network cascade for face detection. In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2015. p. 5325–5334.

MARQUES, Jane A. Usos e apropriações da internet por crianças e adolescentes: análise comparativa das duas ondas da pesquisa Tic Kids Online Brasil. **Pesquisa sobre o uso da internet por crianças e adolescentes no Brasil: TIC Kids Online Brasil**, p. 70–92, 2014.

MOURA, Pedro. **Child Image detection**. [S.l.: s.n.], jul. 2018. https://github.com/phsmoura/child-image-detection.git.

SITE oficial da documentação do OpenCV. [S.l.: s.n.]. https://docs.opencv.org/4.0.0/. Accessed: 2018-11-03.

SITE oficial do OpenCV. [S.l.: s.n.]. https://opencv.org/. Accessed: 2018-08-03.

TRAN, Lam et al. Privacy-cnh: A framework to detect photo privacy with convolutional neural network using hierarchical features. In: THIRTIETH AAAI Conference on Artificial Intelligence. [S.l.: s.n.], 2016.

VIOLA, Paul A.; JONES, Michael J. Robust Real-Time Face Detection. **International Journal of Computer Vision**, v. 57, p. 137–154, 2004.

XIA, Yu; HUANG, Di; WANG, Yunhong. Detecting smiles of young children via deep transfer learning. In: PROCEEDINGS of the IEEE International Conference on Computer Vision. [S.l.: s.n.], 2017. p. 1673–1681.