Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Trabalho de Graduação em Engenharia de Informação

Validação e Análise de Viabilidade de Algoritmos de Detecção Facial

Gabriel Pitali de Carvalho

Santo André

Dezembro de 2019

Gabriel Pitali de Carvalho

Validação e Análise de Viabilidade de Algoritmos de Detecção Facial

Trabalho de Graduação II apresentado durante a Graduação em Engenharia de Informação, como parte dos requisitos necessários para a obtenção do Título Bacharel em Engenharia de Informação.

Universidade Federal do ABC

Orientador: André Kazuo Takahata

Santo André

Dezembro de 2019

Sumário

1 Introdução

Reconhecimento facial é uma tarefa trivial para humanos e há décadas tem sido um desafio para visão computacional e aprendizado de máquina, segundo a referência **??**, desde os anos 90 o tema emerge em diferentes conferencias e com o aumento do poder computacional dos dias atuais, sua capacidade se expande muito, fazendo com que tal assunto receba enorme atenção, principalmente devido ao seu grande valor comercial e as mais diversas aplicações possíveis, como verificação de identidade, controle de acesso, segurança, investigação de imagens em bancos de dados, vigilância, entretenimento ou realidade virtual [**??**] [**??**].

O processo de reconhecimento facial de forma automatizada é separado em 4 principais etapas, conforme detalhado no livro **??**, primeiramente deve ser feita a *detecção facial*, que consiste em validar e localizar a existência de alguma face na imagem ou video, a segunda etapa consiste no *alinhamento facial*, para que todas faces da base de dados sigam o mesmo padrão, a terceira etapa é a *extração de características* que permite a obtenção de informação efetiva que será útil na distinção das diferentes faces, a quarta e última etapa consiste na *correspondência de características*, onde as características extraídas anteriormente são comparadas com outras já conhecidas para que sejam identificadas.

Aprofundando o estudo da primeira etapa, de *detecção facial*, a referência **??** indica duas diferentes metodologias, a primeira baseada em características e a segunda baseada em imagens, ambas posteriormente podem ser separadas em diversas técnicas mais específicas, como por exemplo a análise de características por constelação ou a análise de imagens com redes neurais, onde cada técnica espécifica possui seus prós e contras em relação as demais.

1.1 Objetivos e Motivação

Este trabalho tem como objetivo encontrar a melhor forma de atuar sobre a primeira etapa (*detecção facial*) do processo de reconhecimento facial, avaliando o desempenho qualitativo e quantitativo de diferentes metodologias e ferramentas disponíveis e permitindo a rápida identificação de imagens que não possuem uma face, para satisfazer a necessidade descrita a seguir.

Atualmente empresas e órgãos públicos possuem a necessidade de manter cadastros pessoais mas existe grande demanda para que estes cadastros sejam feitos de forma totalmente virtual pela população, pois isso evita o deslocamento de pessoas até os pontos de cadastro e torna todo o processo muito mais ágil. Certos cadastros incluem fotos de

identificação e isto traz a necessidade de uma verificação feita por humanos para validar se a mesma consiste em uma foto de face frontal, conforme é necessário para o cadastro.

A validação citada já ocorre nas empresas e órgãos públicos que precisam coletar documentos de forma virtual e é feita de forma totalmente manual, onde funcionários tem que verificar cada uma das imagens recebidas e muitas vezes se deparam com fotos sem nenhuma face frontal ou sem condições de serem identificadas (desfocadas, por exemplo), que são rejeitadas para que uma nova imagem seja solicitada. Estas imagens claramente inválidas por estarem em desacordo com o padrão esperado (foto de face frontal), poderiam facilmente ser eliminadas por uma filtragem anterior, reduzindo grande parte do trabalho que é feito hoje manualmente.

2 Materiais e Métodos

Para obter melhor entendimento sobre possíveis ferramentas de visão computacional e detecção facial, este capítulo descreve a execução de testes utilizando a linguagem de programação Python e a implementação da ferramenta OpenCV na mesma linguagem.

2.1 OpenCV

A ferramenta OpenCV, que pode ser encontrada no website Github [**??**], é uma biblioteca de código aberto focada na solução de problemas utilizando visão computacional em tempo real, desenvolvida pela Intel e posteriormente pela Itseez, com suporte a múltiplas plataformas e uso gratuito sobre a licença de código aberto BSD. A ferramenta apresenta suporte a frameworks de aprendizado profundo, como TensorFlow, Pytorch e Caffe e contempla tanto funções básicas, para aplicações como processamento de imagem, alteração de cor ou resolução, até aplicações avançadas, como detecção facial, identificação de características e biometria [**??**].

Uma das ferramentas que será utilizada neste trabalho é a função de detecção de faces da ferramenta OpenCV, que utiliza um classificador em cascata baseado características. Esse é um método eficiente para reconhecimento de faces em imagens proposto por Paul Viola and Michael Jones, amplamente conhecido como método Viola-Jones, onde uma função é treinada com muitos exemplos positivos (imagens que contém o objeto a ser detectado) e negativos (imagens que não contém o objeto a ser detectado) e então utilizada para detectar as mesmas características em outras imagens [**??**].

A detecção de faces utilizando OpenCV consiste em duas etapas principais, a primeira consiste no treinamento do modelo, onde são apresentadas diversas imagens já identificadas para que o modelo encontre padrões positivos e negativos. Após o treinamento, a segunda etapa consiste em utilizar o modelo obtido para identificar, em novas imagens, características semelhantes as vistas nas imagens do treinamento. Neste projeto, será utilizado um modelo fornecido em conjunto com a ferramenta OpenCV, já treinado com diversos exemplos de faces frontais.

O procedimento utilizado permite ainda ajuste de parâmetros para a execução do algoritmo de Viola-Jones, neste teste, serão utilizados: o fator de escala, que é o fator pelo qual as dimensões da imagem serão multiplicadas na tentativa de encontrar faces de diferentes tamanhos (quanto menor, maior a chance de encontrar faces) e o número mínimo de vizinhos, que é número mínimo de detecções, após varias iterações, para uma parte da imagem ser considerada uma face (quanto menor, maior a chance de encontrar

faces).

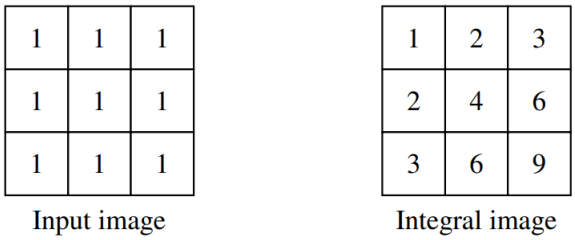
2.2 Algoritmo Viola-Jones

O algoritmo Viola-Jones foi publicado em 2001, no paper "Rapid object detection using a boosted cascade of simple features" [**??**] e é famoso por sua capacidade de detecção de faces com muita velocidade, isso ocorre devido a 3 principais técnicas utilizadas: o cálculo da imagem integral, o algoritmo *AdaBoost* e o classificador em cascata.

2.2.1 Imagem Integral

A primeira etapa do algoritmo Viola-Jones consiste em transformar a imagem original em uma imagem integral, isto é feito calculando o valor de cada pixel como a soma de todos os pixels que estão acima ou a esquerda do mesmo, como ilustrado na figura 1.

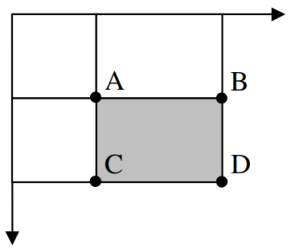
Figura 1 – Imagem original (esquerda) e imagem integral (direita).



Fonte: **??** (**??**)

A utilização desta técnica permite calcular facilmente o tamanho de qualquer retângulo formado entre quatro pixels da imagem, conhecendo apenas o valor dos seus cantos, possibilitando assim a análise rápida de diversas partes da imagem. Tal calculo é feito definindo o retângulo a ser análisado e então aplicando a equação 2.1.

Figura 2 – Representação da área da imagem a ser analisada.

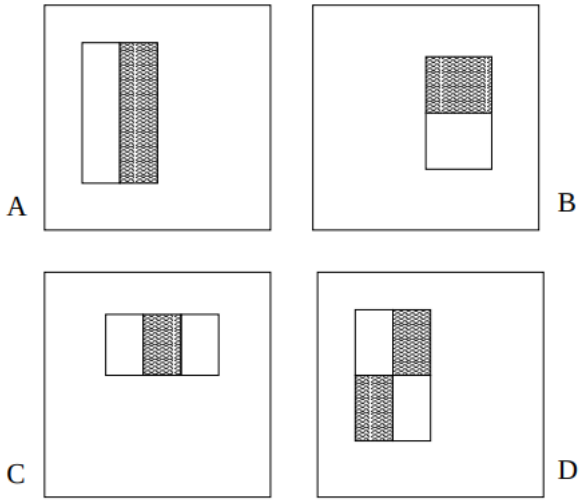


Fonte: **??** (**??**)

$$\text{Soma do retângulo cinza } = D - (B + C) + A \tag{2.1}$$

Com a possibilidade de calcular facilmente a soma dos pixels de um retângulo arbitrário de forma rápida, o algoritmo para detecção pode analisar diversos trechos da imagem, chamados aqui de características, fazendo a comparação de duas ou mais áreas retângulares predefinidas, como os exemplos ilustrado na figura 3.

Figura 3 – Alguns exemplos de características retângulares analisadas.



Fonte: **??** (**??**)

O valor final de cada característica é definido pela soma do valor dos pixels sob o retângulo cinza menos a soma do valor dos pixels sob o retângulo branco.

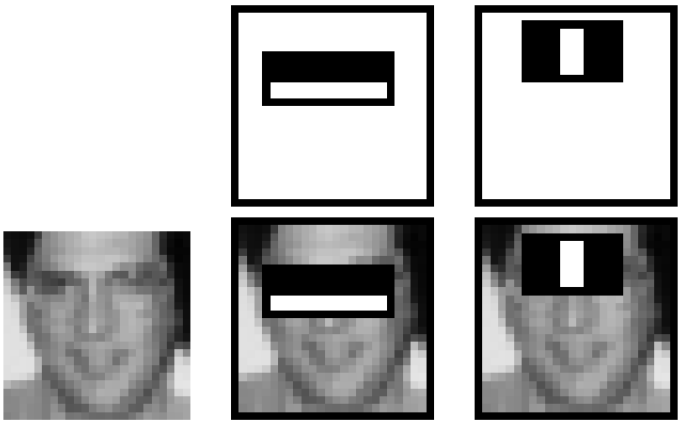
2.2.2 Algoritmo de AdaBoost

As características demonstradas anteriormente, são definidas basicamente como duas ou mais áreas retângulares de qualquer tamanho, tal simplicidade implica na possibilidade da criação de uma enorme variação das mesmas que precisariam ser calculadas diversas vezes, para cada parte de imagem e com diferentes tamanhos, isso implica em um alto custo de processamento, para evitar tal problema, durante a etapa de treinamento do modelo, é utilizado o algoritmo de *AdaBoost* [**??**], que identifica quais são as características com maior probabilidade de acerto.

O *AdaBoost*, que tem seu nome derivado de *adaptative boosting*, é um método de aprendizado de máquina que utiliza a combinação de vários classificadores fracos para obter uma classificação forte, no caso da detecção facial, o algoritmo é utilizado tanto para selecionar um conjunto de características mais eficientes como para treinar o classificador [**??**].

A figura 4 retrata as melhores características registradas por **??** (**??**), fica claro que as mesmas se destacam por evidenciar as regiões dos olhos e do nariz.

Figura 4 – Características retângulares mais eficientes para detecção facial.



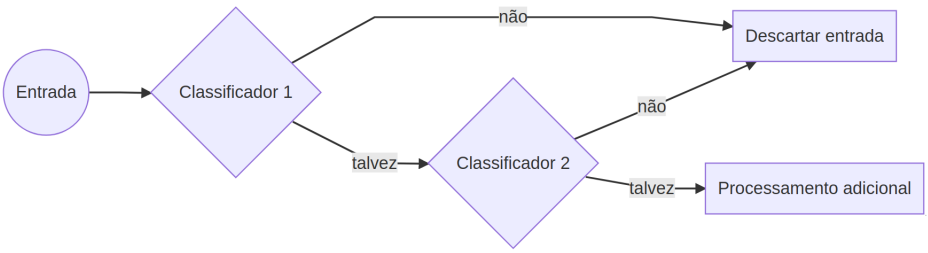
Fonte: ?? (??)

2.2.3 Classificador em Cascata

Pensando que, na maioria dos casos, uma face não ocupa a maior parte de uma imagem a ser identificada, é necessário encontrar uma forma rápida de descartar os elementos do fundo da mesma e concentrar o poder de processamento nos elementos que tem maior probabilidade de serem reconhecidos como uma face, isso leva a uma formulação para o problema onde ao contrário de encontrar faces, é necessário um algoritmo que descarte as "não faces".

Para tal problema, o classificador em cascata apresenta uma ótima solução, esta consiste na utilização de uma série de classificadores que são aplicados de forma sequencial, conforme ilustrado na Figura 5, permitindo que imagens que certamente não possuem faces sejam rapidamente descartadas logo nas primeiras iterações, enquanto imagens com possíveis faces são classificadas por toda cascata, trazendo um elevado nível de confiança ao resultado.

Figura 5 – Diagrama do funcionamento do classificador em cascata.



Fonte: ?? (??)

Um classificador comum, com um único estágio normalmente aceitaria muitos casos de falso negativo, para reduzir a taxa de falsos positivos e de descarte de imagens

relevantes, mas no classificador em cascata, falsos positivos nos primeiros estágios não são um problema, pois serão analisados em outros diversos estágios e provavelmente eliminados.

A utilização desse modelo combinada com o algoritmo *AdaBoost*, possibilita a análise das característica mais eficientes logo no início e consequentemente o descarte muito mais rápido dos casos negativos nos primeiros estágios.

2.3 Conjuntos de imagens para teste

Para analisar a eficiência da implementação do algoritmo *Viola-Jones* na biblioteca *OpenCV*, assim como o seu modelo previamente treinado, foram utilizados dois conjuntos de imagens. Tendo em mente o objetivo deste projeto, de identificar rapidamente imagens que não possuem uma face, são tratadas como imagens positivas as imagens que não possuem faces e como negativas as imagens que possuem ao menos uma face.

O primeiro dataset, nomeado *Hotels-50k* [**??**], será utilizado para representar imagens da classe negativa que consiste em um conjunto de mais de 50 mil imagens de diversos quartos de hotel vazios, pois imagens com características semelhantes a estas são muitas vezes submetidas erroneamente em cadastros pessoais, ao invés de uma imagem da face a ser cadastrada.

O segundo dataset, nomeado *UTKFace*, disponível em [**??**], será utilizado para representar imagens da classe positiva que consiste em um conjunto de mais de 20 mil imagens, com uma única face em cada, de diversas pessoas entre 0 e 116 anos de idade, catalogadas de acordo com idade, raça e sexo. Alguns exemplos das imagens contidas no dataset podem ser vistos na figura 6.

Figura 6 – Exemplos de imagens do dataset *UTKFace*.

Fonte: *UTKFace dataset* [**??**])

Para os testes, foram filtradas apenas imagens de pessoas entre 18 e 60 anos de idade do dataset *UTKFace*, obtendo 17100 imagens e então selecionadas aleatoriamente 4275 imagens do dataset *Hotels-50k*. Assim, foi totalizado um conjunto de 21375 imagens,

onde 80% delas eram positivas (possuíam ao menos uma face) e 20% negativas (não possuíam faces). Tal proporção representa a hipótese de que cerca de 20% de imagens enviadas para cadastros na verdade não possuem uma face.

2.4 Metodologia de Análise

Para melhor analisar os resultados dos testes, foi necessário especificar com clareza como poderiam ser agrupadas as imagens, dada a sua origem e o resultado observado no teste, para isso foram utilizadas as definições da Tabela 1.

Tabela 1 – Grupos observados

Grupo	**Descrição**	**Quantidade**
A	Imagens que contêm uma face	17130
$\overline{A}$	Imagens que não contêm nenhuma face	4275
B	Imagens onde o algoritmo identificou uma ou mais faces	Variável
$\overline{B}$	Imagens onde o algoritmo não identificou nenhuma face	Variável

Definidos os grupos, pode-se utilizar a matriz de confusão da Tabela 2 para facilitar a análise da relação entre os grupos definidos anteriormente. Na Tabela 2 são observados os grupos a (verdadeiro positivo) onde o algoritmo identifica corretamente uma face em cada uma das imagens que realmente contêm uma face, b (falso negativo) onde o algoritmo erroneamente não reconheceu nenhuma face, apesar das imagens conterem uma face cada, c (falso positivo) onde o algoritmo erroneamente identificou ao menos uma face, mesmo as imagens não contendo nenhuma e d (verdadeiro negativo) onde o algoritmo identificou corretamente que não existia nenhuma face nas imagens. É importante destacar que os quatro grupos destacados na matriz de confusão são mutuamente excludentes [**??**].

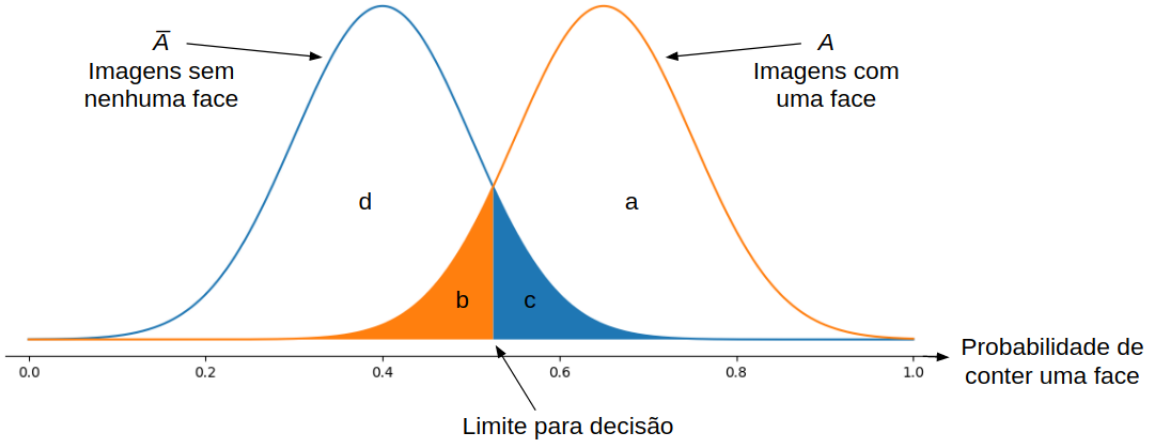
Tabela 2 – Matriz de confusão

	B	$\overline{B}$
A	a (verdadeiro positivo)	b (falso negativo)
$\overline{A}$	c (falso positivo)	d (verdadeiro negativo)

Para melhor entender os agrupamentos da matriz de confusão, pode-se observar na Figura 7 as distribuições que representam a quantidade de imagens dos grupos A e $\overline{A}$ dada a sua probabilidade de conter uma face.

Nas distribuições são destacados os grupos b (falso negativo) e c (falso positivo) e fica evidente que, devido a sobreposição das distribuições A e $\overline{A}$, é necessário definir uma limite para decisão, que pode ser ajustado conforme a necessidade, mas que independente do seu ajuste, sempre existirá um grupo categorizado de forma incorreta.

Figura 7 – Exemplo teórico de uma distribuição normal dos grupos de uma matriz de confusão.

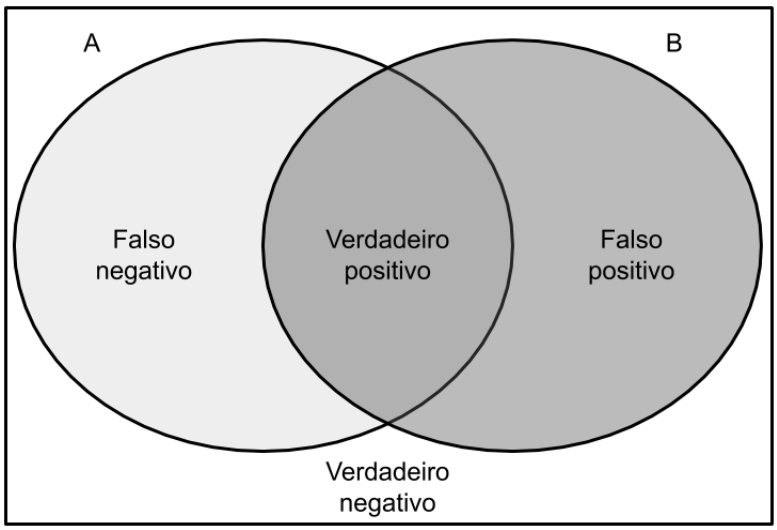


A matriz de confusão também pode ser escrita em termos probabilísticos (Tabela 3), incluindo as probabilidades marginais ou pode ser visualizada no diagrama de Venn correspondente (Figura 8).

Tabela 3 – Matriz de confusão com probabilidades marginais

	B	$\overline{B}$	Soma
A	$P(A \cap B)$	$P(A \cap \overline{B})$	$P(A)$
$\overline{A}$	$P(\overline{A} \cap B)$	$P(\overline{A} \cap \overline{B})$	$P(\overline{A})$
Soma	$P(B)$	$P(\overline{B})$	1

Figura 8 – Diagrama de Venn para as imagens analisadas.



Podem também ser calculadas as medidas tradicionais de sensitividade (a probabilidade condicional do algoritmo identificar ao menos uma face dado que a imagem contém uma face) e especificidade (a probabilidade condicional do algoritmo não identificar

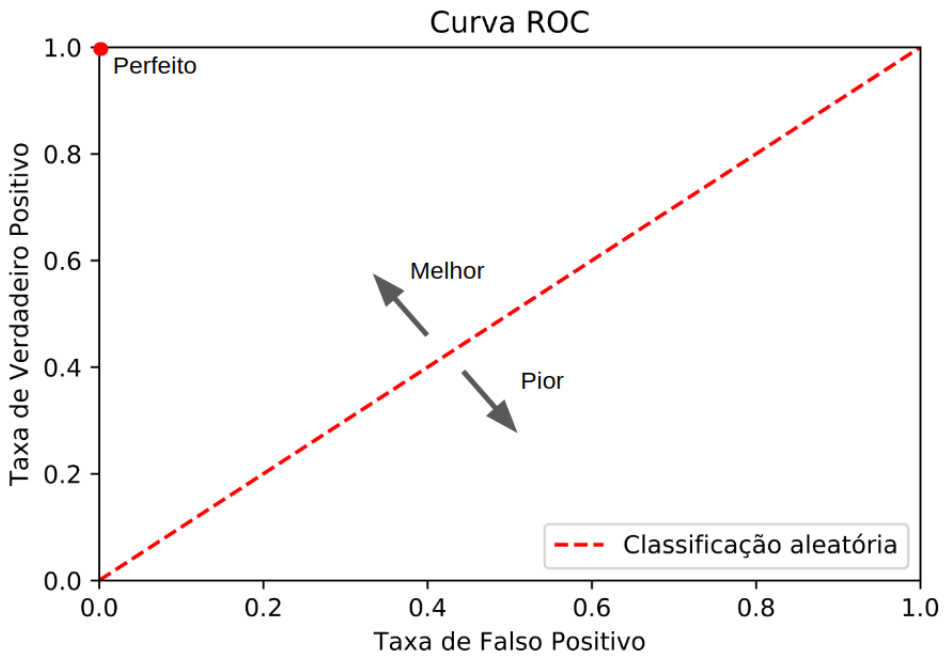
nenhuma face em uma imagem que realmente não contém nenhuma face) conforme as equações 2.2 e 2.3.

$$\text{sensitividade, } P(B|A) = P(A \cap B)/(P(A \cap B) + P(A \cap \overline{B})) = a/(a+b) \tag{2.2}$$

$$\text{especificidade, } P(\overline{B}|\overline{A}) = P(\overline{A} \cap \overline{B})/(P(\overline{A} \cap \overline{B}) + P(\overline{A} \cap B)) = d/(d+c) \tag{2.3}$$

Por fim, a partir da sensitividade e especificidade é possível exibir o resultado do classificador no espaço ROC (*Receiver Operating Characteristic*, traduzido literalmente como Característica operacional do receptor), que consiste em um plano onde o eixo horizontal mede a taxa de falsos positivos (1 - especificidade) e o eixo vertical mede a taxa de verdadeiros positivos (sensitividade), isso permite comparar facilmente diferentes classificadores, pois quanto mais próximo do canto esquerdo superior, melhor a classificação.

Figura 9 – Espaço ROC.



2.5 Análise de Custo

Para entender o valor da utilização de um modelo de reconhecimento facial para filtragem de imagens inválidas, pode-se utilizar o exemplo de uma empresa de cartões de crédito que precisa analisar fotos de clientes antes de fornecer um cartão para os mesmos. Considerando como R o valor presente líquido [**??**], ou seja, a receita total que este cliente traz para empresa, e H como o custo de trabalho humano para a análise de uma imagem, podemos calcular a receita da empresa antes e depois da aplicação de um modelo de detecção facial.

Antes da aplicação do modelo, todas as fotos enviadas deveriam ser analisadas por um humano, gerando o custo de análise manual H, e apenas as fotos que continham uma face trariam a receita R para empresa, relacionando isso aos valores encontrados na matriz de confusão do modelo, onde VP é a taxa de verdadeiros positivos, FP a taxa de falsos positivos, VN a taxa de verdadeiros negativos e FN a taxa de falsos negativos, pode-se obter a equação 2.4. Nesta equação tem-se o $Evento\, A = (VP + FN)$, que representa os casos onde uma foto que realmente contém uma face, multiplicado pela receita de cada cliente R, subtraídos do custo de análise H multiplicado pela quantidade total de fotos recebidas, que equivale a soma de todas as taxas da matriz de confusão.

$$g_0 = (VP + FN) \times R - (VP + FP + FN + VN) \times H \tag{2.4}$$

Após a aplicação do modelo, apenas as fotos que foram classificadas pelo modelo como faces serão analisadas por humanos, gerando custo de análise manual H e apenas as fotos classificadas pelo modelo como face e que realmente contenham uma face se tornarão clientes reais, trazendo receita R para a empresa, com isso, pode-se obter a equação 2.5.

$$g_1 = VP \times R - (VP + FP) \times H \tag{2.5}$$

Para avaliar a receita que o modelo traz para a empresa, pode-se calcular a diferença entre a receita antes e depois do modelo, obtendo a equação 2.6.

$$g_1 - g_0 = VN - (\frac{R}{H} - 1) \times FN \tag{2.6}$$

Considerando que para o modelo trazer valor para a empresa, é necessário que a receita após sua aplicação seja maior que a receita antes da sua aplicação, obtendo a equação 2.7.

$$g_1 - g_0 > 0 \tag{2.7}$$

$$VN - (\frac{R}{H} - 1) \times FN > 0 \tag{2.8}$$

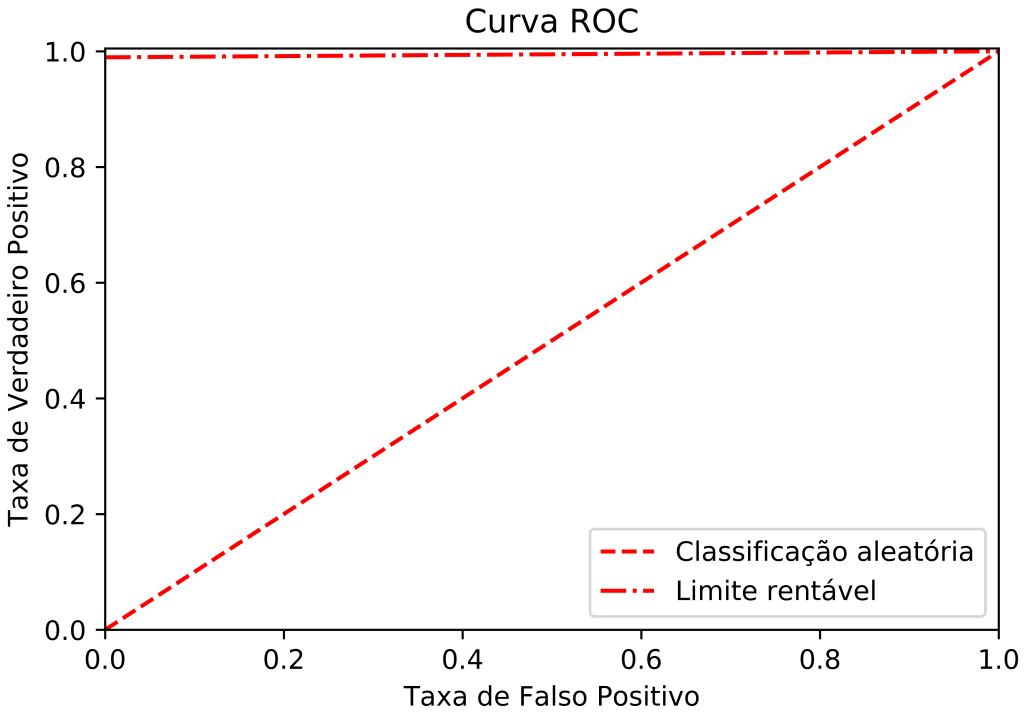
A equação obtida ainda pode ser reescrita em função da taxa de verdadeiros positivos (TVP) e da taxa de falsos positivos (TFP) de forma que seja possível traça-la sobre o espaço ROC e visualizar de forma simples quais classificadores satisfazem o critério necessário para obtenção de receita. Para isso são consideradas a quantidade de imagens positivas ($P = 0.8$) e negativas ($N = 0.2$) definidas anteriormente e as relações 2.9 e 2.10, obtendo por fim a equação 2.11 que pode ser exibida no espaço ROC em 10.

$$FN = P \times (1 - TVP) \tag{2.9}$$

$$VN = N \times (1 - TFP) \tag{2.10}$$

$$TVP = 1 + \left(\frac{N}{P \times \left(\frac{R}{H} - 1 \right)} \times (TFP - 1) \right) \tag{2.11}$$

Figura 10 – Limiar lucrativo exibido no espaço ROC.

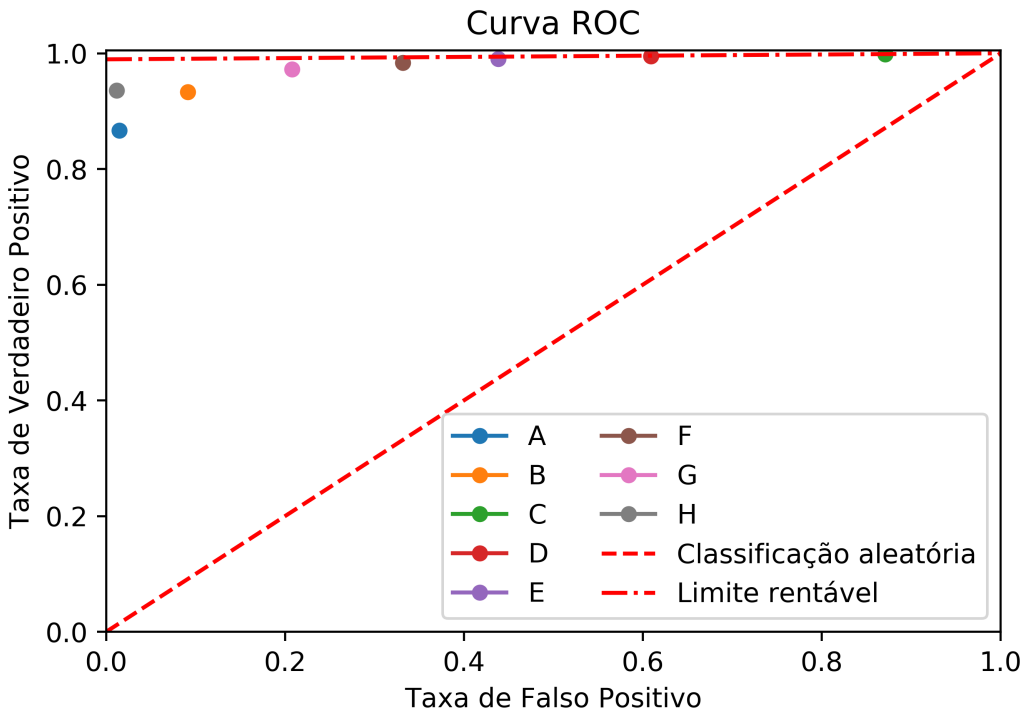


Assim, é possível avaliar em função da matriz de confusão do modelo, do custo de análise manual de um documento, do valor presente líquido de um cliente e da proporção média de imagens positivas e negativas, se a aplicação deste modelo é viável ou não.

3 Resultados e Discussão

Após entender o funcionamento do algoritmo de Viola-Jones e definir a metodologia para análise, foram executadas diversas iterações de classificação sobre o conjunto de imagens, variando os parâmetros de fator de escala para e número mínimo de vizinhos e cada um dos resultados obtidos pode ser exibido como um ponto no espaço ROC obtendo a figura 11, para que todos sejam facilmente comparados e também se torna simples observar se algum dos resultados está posicionado acima o limite lucrativo traçado. Além das classificações utilizando o classificador de Viola-Jones, foi feita uma classificação utilizando o framework Keras (fonte ref) para referência.

Figura 11 – Resultados das classificações sobre o espaço ROC. Os parâmetros utilizados em cada resultado são listados na Tabela 4.



Em nenhuma das tentativas de classificação foi obtido um resultado lucrativo. Aprofundando a análise, pode-se observar os resultados de sensitividade, especificidade e acurácia de cada um dos classificadores na Tabela 5 e a matriz de confusão do resultado que mais se aproximou do cenário lucrativo na Tabela 6, também foi observado que apenas duas imagens de faces não foram reconhecidas, entre as 17100 testadas.

Comparando tal resultado com a análise da equação que define o limiar lucrativo da classificação, é possível perceber que cada imagem positiva que é classificada como negativa gera um prejuízo muito elevado, que precisa ser compensado com a classificação correta

Tabela 4 – Identificação dos resultados da figura 11.

Letra	Método	Fator de Escala	Número Mínimo de Vizinhos
A	OpenCV	1.30	5
B	OpenCV	1.30	2
C	OpenCV	1.05	0
D	OpenCV	1.05	1
E	OpenCV	1.05	2
F	OpenCV	1.05	3
G	OpenCV	1.05	5
H	Keras		

Tabela 5 – Valores de sensitividade, especificidade e acurácia de cada um dos classificadores.

Letra	Sensitividade	Especificidade	Acurácia
A	0.866257	0.985497	0.890105
B	0.933450	0.909006	0.928561
C	0.998480	0.129123	0.824608
D	0.994795	0.390877	0.874012
E	0.990175	0.561871	0.904515
F	0.983509	0.668304	0.920468
G	0.972690	0.792281	0.936608
H	0.936023	0.988538	0.946526

Tabela 6 – Matriz de confusão com os resultados do classificador que mais se aproximou do cenário lucrativo, utilizando os parâmetros fator de escala = 1.05 e número mínimo de vizinhos = 0.

	B	$\overline{B}$	Soma
A	79.8784%	0.1216%	80%
$\overline{A}$	17.4175%	2.5825%	20%
Soma	97.2959%	2.7041%	100%

de uma grande quantidade de imagens negativas. Para o classificador estudado, se torna extremamente difícil atingir tal resultado, mesmo com grande aumento da sensibilidade, classificando quase todas imagens como positivas.

Como uma tentativa extra e com objetivo de comparação, foi utilizado um detector de pele, que se baseia nas cores da imagem, de forma que fosse calculada a quantidade de pele existente em cada imagem, os resultados foram ordenados e então foi traçada a curva ROC na figura 12 que representa os possíveis limiares da quantidade de pele existente em uma imagem que faria a mesma ser considera como face.

A hipótese de que um detector de pele traria bons resultados está relacionada a necessidade de obter um número muito baixo de falsos negativos e a altíssima sensibilidade que é possível obter com tal detector, apesar disso, a quantidade elevada de falsos positivos prejudica bastante os resultados, obtendo um classificador quase igual ao aleatório.

Figura 12 – Curva ROC para o detector de pele.