

# DESENVOLVIMENTO DE UM PROTÓTIPO DE SOFTWARE DE RECONHECIMENTO FACIAL: ESTUDO DO SISTEMA PRESENTE!!!

**PERAMO, Luiz Fernando**

Faculdade de Tecnologia de Mogi Mirim - FATEC  
*lperamo@yahoo.com.br*

**MACEDO, Paulo Cesar de**

Faculdade Santa Lúcia  
*paulo.macedo@fatec.sp.gov.br*

**MORAES, Marcos Roberto de**

Faculdade Santa Lúcia  
*professormoraes@gmail.com*

## RESUMO

*Visando a responsabilidade de uma Instituição de Ensino (IE) no controle de presença dos alunos, que atualmente ocorre por meio de uma confirmação verbal ao professor, considerando uma questão de segurança e trazendo os pais ou parentes mais próximos da vida escolar do aluno, foi idealizado o sistema PRESENTE!!! O objetivo principal deste artigo é desenvolver um sistema de reconhecimento facial sem custo para identificar o aluno e gerenciar sua frequência às aulas, disparando um comunicado aos interessados caso este não compareça à aula após o tempo limite de atraso, que pode ser previamente configurado pela direção ou coordenação. Para este sistema, propõe-se utilizar algumas bibliotecas que trazem em seu conteúdo algoritmos de reconhecimentos e aprendizagem de máquinas. Para facilitar o processo de desenvolvimento escolheu-se a linguagem Python, que é uma linguagem de alto nível, porem de baixa complexidade de desenvolvimento, com uma vasta gama de biblioteca, tornando-a multiplataforma. Foram realizados alguns testes individuais da ferramenta e um teste maior em um evento.*

*Estiveram presentes cerca de 750 pessoas entre visitantes e alunos regulares da instituição. O algoritmo implementado mostrou-se 85% de precisão reconhecendo a maioria dos alunos presentes e dispensando a tradicional chamada feita pelos professores.*

**PALAVRAS-CHAVE:** *Biometria; Frequência escolar; Reconhecimento Facial; Aprendizado de Máquina.*

## INTRODUÇÃO

O reconhecimento facial é um dos processos de identificação mais utilizado pelos seres humanos, pois permite identificar rapidamente qualquer pessoa. Experimentos mostram que até um bebê de três dias de idade é capaz de distinguir entre rostos conhecidos, embora o reconhecimento facial seja uma tarefa simples para os seres humanos, é extremamente complexa para as máquinas (TURATI *et al.*, 2006).

As características internas (olhos, nariz e boca) e as características externas (forma da cabeça e traços essenciais da face) são usadas para reconhecimento de rosto. Mas como nosso cérebro a codifica? Nosso cérebro possui células nervosas especializadas que respondem às normas locais de uma cena, como linhas, bordas, ângulos ou movimentos. Como não vemos o mundo como peças dispersas, nosso córtex visual se desenvolve com diferentes fontes de informação em padrões úteis (KANADE, 1973).

As características geométricas de um rosto são uma abordagem mais intuitiva para o reconhecimento facial. Um dos primeiros sistemas automatizados de reconhecimento facial foi descrito em pontos de marcador (posição dos olhos, orelhas e nariz) e foram usados para construir um vetor de características (distância entre pontos, ângulo entre eles, entre outros). O reconhecimento foi elaborado por meio da distância euclidiana (distância entre dois pontos) entre vetores característicos de uma sonda e imagem de referência. Este método é contraído por sua própria natureza, mas existe muita diversidade: o registro precisa dos pontos de divulgação, independentemente dos algoritmos (BRUNELLI; POGGIO, 1992).

Para Turk e Pentland (1991), o método *Eigenfaces* tomou uma abordagem holística para o reconhecimento facial: uma imagem do rosto é um ponto de um espaço de imagem de alta dimensão e uma representação de dimensão inferior encontrada, onde a classificação é mais fácil. O subespaço menor é encontrado com uma Análise de Componentes Principais (ACP),

que identifica os eixos com variação máxima, embora este tipo de transformação seja otimizada do ponto de vista da reconstrução, nenhum rótulo de classe é questionado. Os eixos com variação máxima não são necessários para qualquer informação discriminatória. Então, uma projeção específica da classe com uma Análise de Discriminação Linear (ADL) é aplicada para reconhecimento facial. Uma ideia básica é minimizar uma variação dentro de uma classe, ao mesmo tempo em que maximiza uma variação entre as classes.

O presente artigo fez uso de tecnologias no sentido de aprimorar o reconhecimento facial como ferramenta de apoio à identificação de pessoas e propõe a criação de um protótipo de *software* para o reconhecimento de características faciais de estudantes durante a entrada e saída de uma IE. Sua aplicação destina-se ao controle de frequências escolares.

O protótipo de *software* apresentado neste artigo foi projetado para ser uma ferramenta simples e de fácil utilização. Entre as ferramentas de desenvolvimento utilizadas, considera-se o uso do sistema operacional Linux, da linguagem de programação Python, do *software* de desenvolvimento de banco de dados SQLite e das bibliotecas de visão computacional OpenCV e Dlib. Em relação ao *hardware* utilizado, destaca-se o uso do dispositivo de uma câmera tipo *webcam* responsável pela captura de imagens.

O artigo propõe um modelo de *software* para auxiliar na captura de dados fornecidos pela câmera, gerando uma base de dados de imagem local, que será utilizada no reconhecimento facial dos estudantes. Como resultado, é verificada a performance do algoritmo no processo de detecção e reconhecimento dos estudantes presentes na cena a partir de dados obtidos pela câmera. A validação destes dados é feita por meio de testes práticos utilizando o protótipo de *software*, que demonstram a viabilidade do sistema.

Todos os dados utilizados neste estudo foram coletados a partir do modelo desenvolvido, sendo os experimentos realizados com pessoas em um determinado ambiente previamente definido.

Quanto à análise dos dados, há a necessidade de armazenamento dos dados que serão coletados pelo algoritmo para posterior análise das características da face dos estudantes. O algoritmo desenvolvido juntamente de sua base de dados implementada fará a coleta e o armazenamento de imagens da face.

O presente artigo foi fundamentado sob a ótica motivacional de desenvolvimento de um protótipo de *software* de presença aplicado ao uso da câmera. Dessa forma, tem-se como principal motivação o atendimento à necessidade de desenvolvimento de um algoritmo de identificação e reconhecimento facial para a marcação de presença de alunos em uma escola privada. A implementação desse sistema baseado nos padrões de identificação da face

favorecerá o processo de gestão permitindo a identificação de pessoas quanto às suas características, sendo um estímulo para o desenvolvimento de uma base centralizada das imagens de face, o que possibilita o registro da presença devidamente cadastradas no sistema a partir da identificação das imagens faciais.

Para viabilizar o desenvolvimento dessa aplicação, o uso da câmera foi adotado como alternativa viável para construção do protótipo. Ela deverá ser instalada em um local específico no ambiente, atuando como um mecanismo de coleta de dados e auxiliando no processo de captura de imagens e informações. É esperado que tenha um reconhecimento maior que 75%.

Outro ponto importante é que Machado (2016) concluiu que em média 20% do tempo é desperdiçado com trabalhos burocráticos, entre eles está a chamada de presença de alunos. Isso representa a perda de um dia na semana para tais atividades, e é claro, a perda de aquisição de conhecimento, que não tem como mensurar.

Esse artigo então visa de maneira específica a automatização do controle de presença, para minimizar esse tempo gasto com a chamada de presença em sala de aula.

## **2. FUNDAMENTAÇÃO TEÓRICA**

Nesse capítulo, destaca-se como os computadores lidam com as imagens após sua captura, bem como o reconhecimento e a linguagem de programação adequada.

### **2.1 Processamento de Imagem**

Tornando-se parte do cotidiano do homem, os computadores têm se tornado cada vez mais eficiente no auxílio de funções e tarefas visuais antes executadas pelo ser humano. Percebe-se um crescimento muito acelerado em tarefas que eram tidas como impossíveis de serem realizadas por computadores, como reconhecimento de padrões, cores e imagens (GOMES; VELHO, 2002).

De acordo com Siqueira (2010), ao associarmos a capacidade computacional de processamento e reconhecimento de imagens capturadas, temos a possibilidade de reproduzir tarefas repetitivas e onerosas de tempo e trabalho com grande êxito e qualidade. Estas situações, tais como, exames médicos por imagem ou imagens de satélite por exemplo, podem resultar em novas imagens, dados e coordenadas.

Para Siqueira (2010) um dos maiores e mais importante uso do reconhecimento de dados em imagens é, por exemplo, na área de saúde,

por auxiliar no diagnóstico precoce de doenças identificadas por meio de exames de radiografia ou tomografia computadorizada.

Segundo Gomes e Velho (2002) outras situações fazem com que a disseminação desta tecnologia seja cada vez maior, a utilização de amostras captadas por meio de satélites torna possível a previsão do clima e do tempo em situações que podem salvar centenas de vidas, como no caso de furacões e tornados.

Citado por Gomes e Velho (2002) o sistema de processamento digital de imagens admite um conjunto de imagens de entrada que retorna um conjunto de imagens de saída de forma adequada às necessidades do problema abordado.

A melhoria da informação visual para a interpretação humana e o processamento de dados de cenas para a percepção automática por meio de máquinas, sendo o processamento dessas imagens realizado por meio de *software* específicos são dois dos principais métodos de processamento de imagens digitais (GONZALEZ; WOODS, 2009).

Segundo Esquef (2002), as máquinas obtêm informações e características do ambiente a sua volta de forma similar à percepção humana, com o uso de imagens produzidas por diferentes dispositivos físicos como câmeras digitais, ressonância magnética, entre outros. A produção e utilização de imagens podem favorecer o reconhecimento de objetos, assim como o reconhecimento de pessoas.

Para Gomes e Velho (2002), um sistema de processamento digital de imagens admite um conjunto de imagens de entrada e retorna um conjunto de imagens de saída de forma adequada às necessidades do problema abordado.

Subdividido em algumas etapas o processamento digital de imagens pode ser, iniciando-se pela formação e digitalização da imagem, pré-processamento, segmentação, pós-processamento e finalmente pela extração de atributos, medidas, classificação e reconhecimento (ESQUEF, 2002).

Ainda de acordo com Esquef (2002), a segmentação é considerada a etapa mais crítica de todo o processo, pois a imagem é dividida em diferentes regiões, que serão posteriormente utilizadas para a identificação e extração de informações relevantes. Essa característica faz da segmentação um ponto crítico, pois quaisquer erros ou distorções ocorridas neste processo são transmitidos às demais etapas, intensificando a possibilidade de obtenção de resultados indesejados. Posterior à etapa de segmentação, é realizada a etapa de reconhecimento que consiste na classificação dos objetos segmentados a partir de informações extraídas da própria imagem.

A finalidade da análise de imagens, seja por um observador humano ou por uma máquina, é extrair informações úteis e relevantes para cada aplicação desejada.

## 2.2 Machine Learning

Segundo VanderPlas (2016) *Machine Learning* ou aprendizado de máquina é o desenvolvimento de técnicas para ensinar as máquinas a executarem tarefas de forma automática prevendo um resultado de acordo com os parâmetros informados.

Esse sistema de aprendizado é um programa que toma decisões baseado em experiências acumuladas por meio de soluções bem-sucedidas anteriormente.

De acordo VanderPlas (2016) existem dois tipos de aprendizado: Supervisionado e Não-Supervisionado.

VanderPlas (2016) considera que o aprendizado supervisionado é quando temos um resultado esperado para a nossa previsão, o programa é treinado com um conjunto de dados pré-definidos e toma decisões precisas quando recebe novos dados, enquanto o aprendizado não supervisionado não tem um resultado esperado ou uma resposta certa, por exemplo: melhor rota para um carro chegar ao seu destino.

## 2.3 Reconhecimento Facial

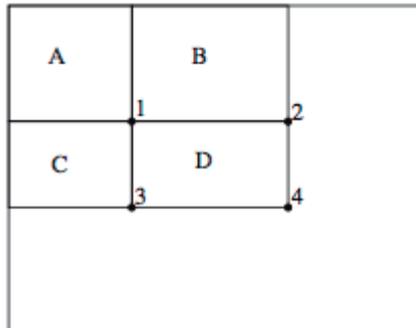
Para iniciarmos o estudo sobre Reconhecimento facial precisamos entender o funcionamento do algoritmo de Viola e Jones (2001), baseado em filtros de *Haar* em cascata, qual localiza as features ou características de uma imagem.

Segundo Viola e Jones (2001), seu método possui três contribuições principais que o distingue de outros algoritmos e o torna computacionalmente mais eficiente, elas são: a imagem integral, o algoritmo de aprendizado baseado em *AdaBoost* e um método para combinar classificadores de complexidade crescente, baseados em *Haar-like features* para criar um filtro em cascata eficiente.

Como mostra a **Figura 1**, o valor da imagem integral na posição 1 é a soma dos pixels do retângulo A. O valor referente à posição 2 é igual à soma de A+B, para a posição 3 é A+C e para a posição 4 é A+B+C+D. Logo, a soma dos valores dos pixels do retângulo D pode ser calculada por  $4+1-(2+3)$ .

A detecção de objetos usando classificadores de cascata baseados em recursos de *Haar* é um método eficaz de detecção de objetos proposto por Viola e Jones (2001). É uma abordagem baseada em aprendizado de máquina em que é treinada a partir de muitas imagens. Ela é usada para detectar objetos em outras imagens.

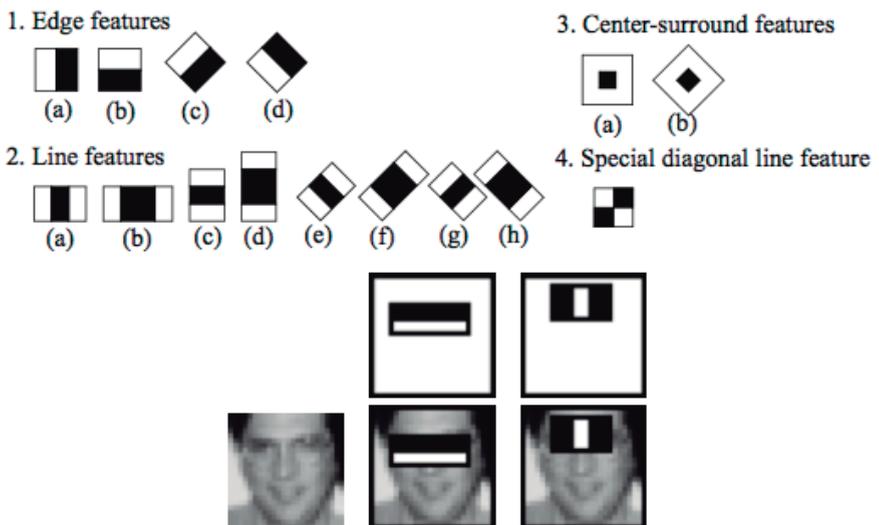
**Figura 1** - Classificador de complexidade crescente



**Fonte:** Viola e Jones (2001, p 3)

Inicialmente, o algoritmo precisa de muitas imagens positivas (imagens de faces) e imagens negativas (imagens sem faces) para treinar o classificador, então precisamos extrair recursos dele. Para isso, são usados os recursos de arremesso exibidos na **Figura 2**. Cada recurso é um único valor obtido subtraindo a soma de *pixels* sob o retângulo branco da soma de *pixels* sob o retângulo preto (VIOLA; JONES, 2001).

**Figura 2** - Exemplos de *haar-like features* e sobreposição em uma face



**Fonte:** Viola e Jones (2001, p 4)

De acordo com Viola e Jones (2001) o algoritmo de aprendizado baseado em *AdaBoost* define quais das diversas *features* existentes serão utilizadas. O que foi observado é que um número pequeno de *features* combinadas podem formar bons classificadores, sendo o principal problema escolher quais *features* são as mais adequadas. Para isso, a etapa de aprendizado realizada por uma variante do *AdaBoost* é responsável pela escolha de uma *feature* para treinar cada classificador para que ele consiga separar, da melhor maneira possível, amostras positivas e negativas. Assim, para cada *feature*, o algoritmo determina um limiar que minimize o número de amostras classificadas erroneamente. Na prática, ainda segundo Viola e Jones (2001), nenhuma *feature* consegue uma baixa taxa de erro nesse processo de classificação. Para cada ciclo do processo de *boosting*, que deve escolher uma *feature* dentre 180.000 possíveis, as selecionadas nos primeiros ciclos possuem taxa de erro de 0,1 a 0,3, já as selecionadas durante os ciclos finais chegam a taxas de erro de 0,4 e 0,5.

Para Viola e Jones (2001), o produto do processo anterior, os classificadores fracos, cada um contendo uma *feature* e um limiar, são então combinados com o auxílio de pesos para formar um classificador forte com o intuito de criar uma estrutura que possa minimizar a ocorrência de falsos negativos. Esses são organizados no formato de uma árvore de decisão degenerada que consiste no detector em cascata.

Segundo Viola e Jones (2001), o funcionamento básico de uma estrutura como essa ocorre pela passagem de todas as sub janelas de uma imagem por diversos classificadores, sendo os primeiros mais simples que os seguintes. Se é obtido um resultado positivo pela análise do primeiro classificador, a janela segue para o seguinte, que se também for aprovada, continua pelos próximos classificadores até ser rejeitada ou então chegar ao final da cascata. Em qualquer uma das etapas, uma janela pode ser rejeitada, isso quer dizer que o detector entende que ali uma face não está presente.

De acordo com Viola e Jones (2001), a estrutura dessa cascata reflete uma característica particular de um processo de detecção que consiste na existência de uma quantidade muito maior de sub janelas negativas em uma imagem do que sub janelas positivas, ou seja, que contenham faces. Assim, tem-se como objetivo rejeitar o maior número de janelas negativas logo nas primeiras etapas fazendo com que apenas janelas positivas continuem pelo processo de detecção. Isso é interessante, pois quanto mais se caminha pela estrutura em cascata, passa-se por classificadores mais complexos, e que, portanto, exigem um custo maior de processamento. Logo, quanto antes uma janela negativa é rejeitada mais eficiente se torna o processo,

pois requisita que apenas janelas positivas, que é um evento raro, passem por toda a árvore.

Para Viola e Jones (2001), em cada etapa os classificadores se tornam mais custosos, pois a medida que se percorre a árvore eles são treinados usando os exemplos que passaram por todas as etapas anteriores. Consequentemente, os classificadores que surgem posteriormente se deparam com uma tarefa de classificação mais difícil. Percebe-se que o treino de uma cascata envolve um dilema entre a eficiência computacional e desempenho do detector. Isso ocorre porque, em geral, classificadores com mais *features* alcançam taxas maiores para detecção e menores para falsos positivos. Entretanto, classificadores como esses requerem maior tempo de processamento. Chega-se à conclusão que, para otimizar um processo como esse, deve-se atentar para o número de etapas de classificadores, o número de *features* e o limiar em cada uma dessas etapas. Como essa otimização é um processo extremamente complicado, na prática, o que se faz é criar uma cascata que a cada etapa reduza a taxa de falsos positivos e a taxa de detecção. Para cada etapa, define-se uma meta para o valor mínimo de redução de falsos positivos e para o valor máximo de redução para a detecção. Então, acrescentam-se *features* até que se atinja as metas estabelecidas para cada etapa e etapas são adicionadas até que se alcance uma taxa de falso positivos e de detecção desejadas para a cascata.

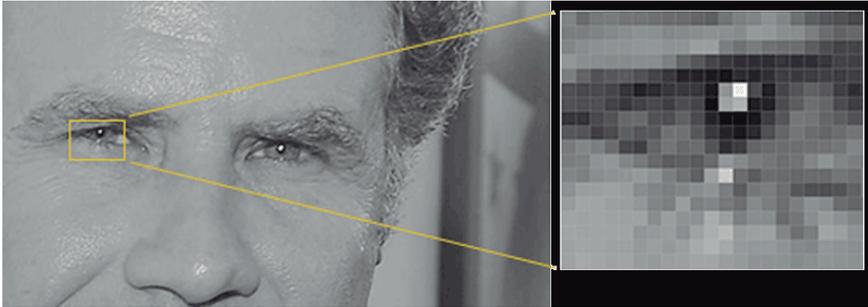
Segundo Blanc-Talon *et al.* (2017), a utilização de *Deep Learning* (*Deep learning* é um tipo de *machine learning* que treina computadores para realizar tarefas como seres humanos, o que inclui reconhecimento de fala, identificação de imagem e previsões )no reconhecimento facial traz a segurança no reconhecimento de faces, tendo até 98% de acertos.

Como visto anteriormente a detecção de rostos se tornou popular no início dos anos 2000, quando Viola e Jones (2001) inventaram uma maneira de detectar rostos que eram rápidos o suficiente para rodar em câmeras baratas. No entanto, soluções muito mais confiáveis existem agora como descrito por Blanc-Talon *et al.* (2017). Mas para isso temos como base o método inventado em 2005 chamado Histograma de Gradientes Orientados, ou apenas HOG.

Para encontrar rostos em uma imagem, o primeiro passo é tornar uma imagem em preto e branco como no método de Viola e Jones (2001), porque não é preciso cores para encontrar rosto.

Na **Figura 4** pode-se notar o próximo passo que é olhar cada *pixel* da imagem, um de cada vez. Para cada *pixel*, é necessário ver os demais (*pixels*) que o rodeiam diretamente:

**Figura 4** - Comparação entre pixel e seu vizinho imediato.



**Fonte:** Geitgey (2017, p.2)

O objetivo é descobrir quão escuro o *pixel* atual é comparado aos *pixels* que o rodeiam diretamente. Então destaca-se com uma linha indicativa conforme na **Figura 5**, mostrando em qual direção a imagem está ficando mais escura:

**Figura 5** - Destacando o sentido do mais escuro



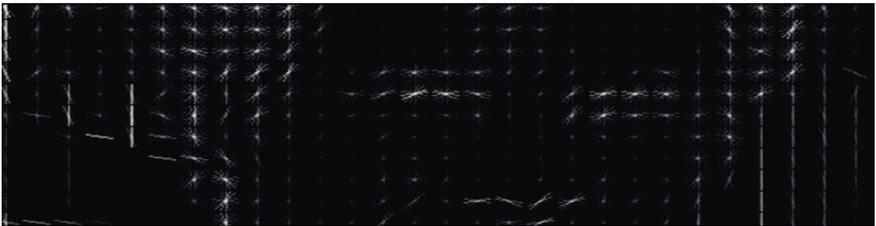
**Fonte:** Geitgey (2017, p. 2)

Segundo Blanc-Talon *et al.* (2017) se repetir esse processo para cada *pixel* da imagem, será substituído cada *pixel* por uma linha (seta). Essas são chamadas de gradientes e mostram o fluxo da luz para a escuridão em toda a imagem.

Ainda de acordo com Blanc-Talon *et al.* (2017), salvar o gradiente para cada *pixel* dá muitos detalhes e aumenta a necessidade de processamento robusto, para ter um processamento mais eficaz e rápido, a imagem é dividida em pequenos quadrados de 16x16 *pixels* cada. Em cada quadrado,

é contado quantos gradientes apontam em cada direção principal (quantos apontam para cima, apontam para baixo, para a direita, para a esquerda, etc.). Então, é substituído esse quadrado na imagem pelas direções das linhas indicativas (setas) mais fortes como na **Figura 6**. O resultado é a transformação da imagem original em uma representação muito simples que captura a estrutura básica de um rosto de uma maneira simples:

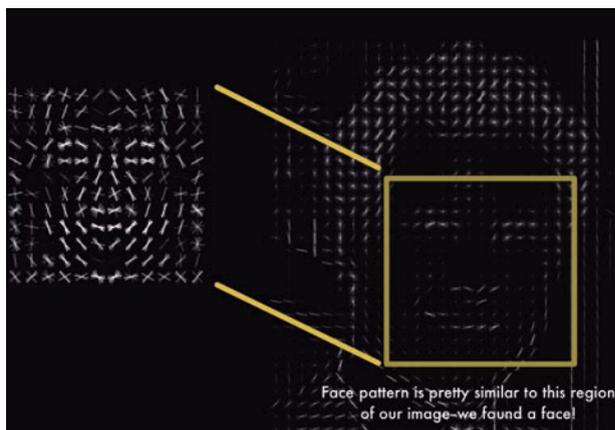
**Figura 6** - Contando as setas e deixando mais forte



**Fonte:** Geitgey (2017, p.2)

Baseado em *Blanc-Talon et al.* (2017), para localizar rostos nesta imagem HOG, é preciso encontrar a parte da imagem que parece a mais semelhante à um padrão HOG conhecido que foi extraído de várias outras faces de treinamento, como mostra a **Figura 7**.

**Figura 7** - Comparando com o padrão HOG

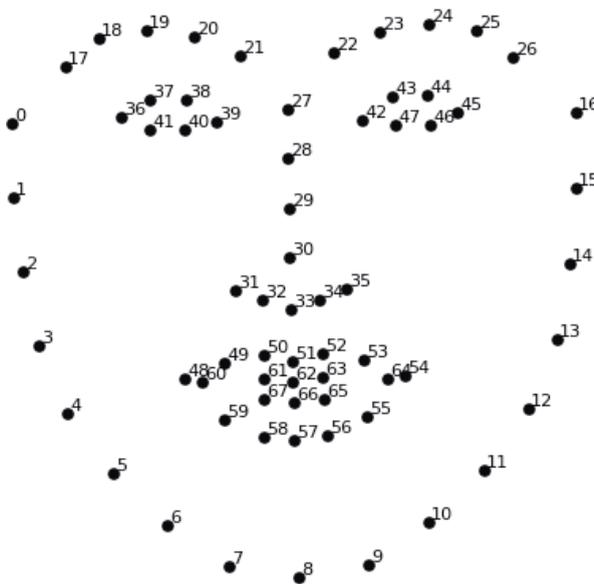


**Fonte:** Geitgey (2017, p.2)

Para Blanc-Talon *et al.* (2017), assim é isolado os rostos na imagem, e para lidar com o problema que os rostos de diferentes direções parecem totalmente diferentes de um computador, é usado um algoritmo chamado estimativa de ponto de referência de face. Há muitas maneiras de fazer isso, e uma delas é a abordagem inventada por Kazemi e Sullivan (2014).

Para Blanc-Talon *et al.* (2017), a ideia básica é chegar aos 68 pontos específicos (chamados pontos de referência) que existem em cada face, a parte superior do queixo, a borda externa de cada olho, a borda interna de cada sobrancelha, etc. Então, é necessário treinar uma máquina com o algoritmo de aprendizagem para encontrar estes 68 pontos específicos em qualquer face, como é observado na **Figura 8**.

**Figura 8** - Pontos de Referências



**Fonte:** Geitgey (2017, p.3)

Segundo Blanc-Talon *et al.* (2017) a imagem é distorcida para que os olhos e a boca fiquem centralizados da melhor forma possível, ilustrado na **Figura 9**. Não é feito qualquer tipo de distorção 3D porque isso introduziria distorções na imagem. São utilizadas apenas transformações básicas de

imagem, como rotação e escala, que preservam as linhas paralelas (chamadas de transformações afins).

**Figura 9** - Alinhando o rosto para realizar a comparação com a face conhecida



**Fonte:** Geitgey (2017, p.3)

As medidas que parecem óbvias para nós seres humanos (como a cor dos olhos) não fazem sentido para um computador olhando para *pixels* individuais em uma imagem. Blanc-Talon *et al.* (2017) descobriram que a abordagem mais precisa é permitir que o computador calcule as medidas para se recompor. A aprendizagem profunda faz um trabalho melhor do que os seres humanos em descobrir quais partes de um rosto são importantes para medir.

Baseado em Blanc-Talon *et al.* (2017), a solução é treinar uma Rede Neural Profunda (RNP). Então, é preciso executar as imagens faciais através da rede pré-treinada para obter as 68 medições para cada face.

## 2.4 Phyton

De acordo com Coelho (2007), Phyton foi criada em 1991 pelo matemático holandês Guido van Rossum, é uma linguagem de altíssimo nível e foi criada e estruturada para o ensino da programação, por isso ela é de fácil compreensão e estrutura. O ideal do seu criador é Programação de Computadores para todos.

Ainda segundo Coelho (2007), a linguagem de programação Phyton é uma linguagem orientada a objetos, clara, objetiva e poderosa, podendo ser usada para desenvolver sistemas e grandes projetos e oferece suporte à integração com outras linguagens e ferramentas.

Para Menezes (2010), Phyton é um *software* livre com linguagem completa, contém muitas bibliotecas para acessar banco de dados, processar

arquivos XML, construir interfaces gráficas e até jogos. Isso aumenta a produtividade do programador, reduz o número de erros pois, ao utilizarmos bibliotecas, usamos programas desenvolvidos e testados por outros programadores.

Uma grande vantagem de Python é a legibilidade dos programas escritos nessa linguagem, não se usa ponto (.), ponto e vírgula (;), marcadores ({} ) e pode ser instalado em qualquer plataforma computacional (LABAKI, 2017). Percebe-se então que essa linguagem é mais adequada para implementação da solução proposta nesse artigo científico.

### 3. MÉTODO

A cada ano, pais e responsáveis de crianças e adolescentes passam pelo dilema de decidirem que escola seus filhos frequentarão. Talvez por isso, escolham a escola X pela excelente reputação acadêmica ou pelos professores altamente qualificados ou pelas atividades extracurriculares. No entanto, é provável que a segurança de seus filhos esteja entre as principais razões por terem escolhido matriculá-los na escola Y, demonstrando assim sua confiança nela.

Visando a responsabilidade da escola no controle de presença dos alunos, onde hoje ocorre por meio de uma confirmação verbal ao professor, considerando uma questão de segurança e trazendo os pais mais próximos a vida escolar do aluno, idealizamos o sistema PRESENTE!!!.

Utilizando o reconhecimento facial para identificar os alunos e confirmar a presença na escola, trazendo assim a segurança aos pais ou responsáveis que seu filho está realmente na escola e para qualquer coisa que aconteça, serão comunicados. O professor terá um tempo a mais para trabalhar com os alunos sem se preocupar com a parte burocrática da chamada. Caso o aluno não passe pela câmera, respeitando o tempo limite de atraso que pode ser configurado pela direção ou coordenação, o sistema dispara um comunicado aos pais ou responsáveis.

Para este sistema, propomos desenvolver para reconhecimento facial individual, para serem instalados em totens, em locais determinados pela direção da escola.

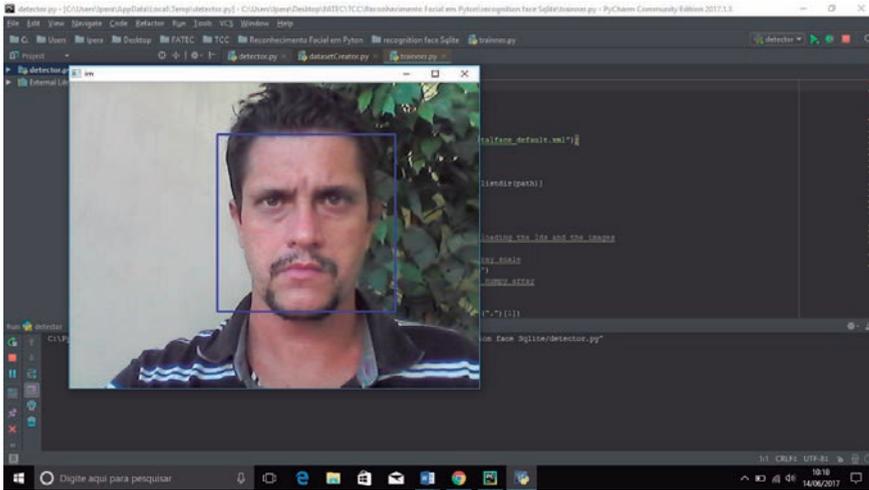
Para trabalhos futuros, poderá ser elaborado um reconhecimento em multifaceis, ou seja, uma única câmera reconhece todos a longa distância, podendo ser instalado na entrada dos alunos, não sendo necessário ficarem estáticos em frente da câmera, sendo capaz de serem reconhecidos em movimento e em grupos.



funcionamento. Então para este projeto decidiu-se pela utilização do *SQLite*.

Na **Figura 11** percebe-se como a foto é capturada pela tela do detector e armazenada no banco de imagens.

**Figura 11** - Tela do detector de 20×20 ou 24×24 pixels.



## b) Módulo *Recognition*:

Para o *Recognition* foi utilizado a ferramenta de reconhecimento de faces do *Dlib*.

Esta ferramenta mapeia uma imagem de um rosto humano para um espaço vetorial de 128 dimensões onde as imagens de uma mesma pessoa estão próximas umas das outras e imagens de pessoas diferentes estão muito distantes. Portanto, você pode executar o reconhecimento facial mapeando as faces em 128 distâncias diferentes e, em seguida, verificar se a distância euclidiana (distância entre dois pontos) é pequena o suficiente.

Ao usar um limite de distância de 0.6, o modelo *Dlib* obtém uma precisão de 99,38% no *benchmark* padrão de reconhecimento facial, que é comparável a outros métodos de última geração para reconhecimento de rosto. Essa precisão é apresentada com um par de imagens. (DLIB, 2018).

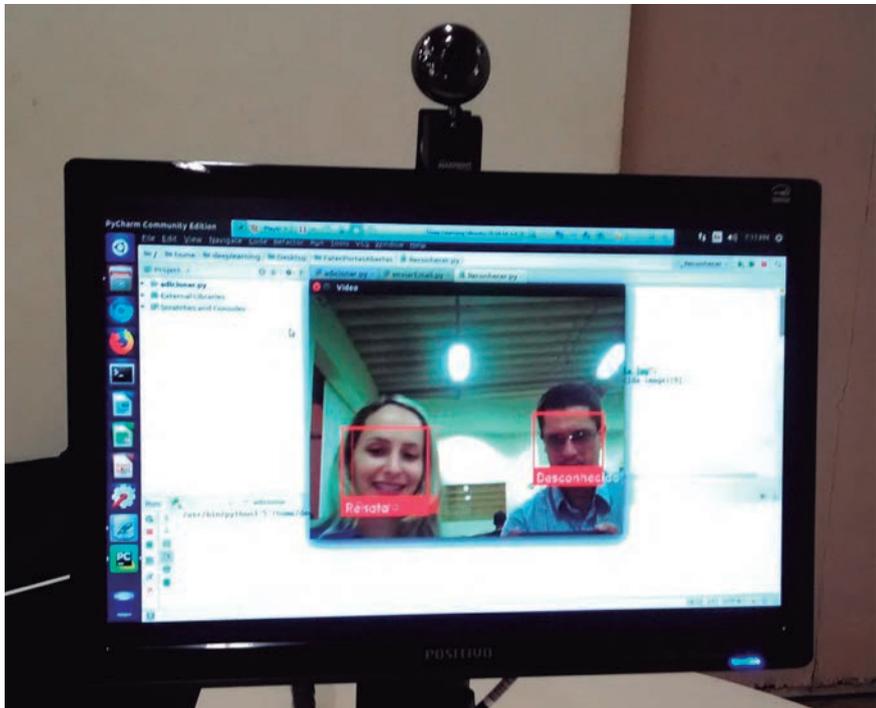
O reconhecimento da *Dlib* codifica uma imagem usando o algoritmo HOG para criar uma versão simplificada da imagem. Usando esta imagem simplificada, encontra a parte da imagem que mais parece uma codificação HOG genérica de uma face e, descobre a pose do rosto, encontrando os

principais marcos no rosto. Assim que encontra esses marcos, usa-os para deformar a imagem de modo que os olhos e a boca estejam centralizados.

Processa a imagem da face centralizada por meio de uma rede neural que mede os traços do rosto e salva essas 128 medições de vetores. Verifica todos os rostos medidos no passado e confirma qual imagem tem as medidas mais próximas das medidas do rosto, conforme a **Figura 12** a seguir.

Para melhorar o processamento, passa a escala da face para  $\frac{1}{4}$  após encontrar uma face relacionada, aumenta a escala novamente em  $\frac{1}{4}$  e compara a face no total.

**Figura 12** - Tela do Recognition.



Após identificado a face do aluno, o sistema Presente!!! envia um comunicado via *e-mail* com data e hora para o responsável. Isto ocorre tanto na chegada, quanto na saída do aluno.

Na **Figura 13** um exemplo do *e-mail* que o responsável receberá no momento do reconhecimento.

**Figura 13** - Exemplo de *e-mail* recebido.



## CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi desenvolver um protótipo de um sistema de reconhecimento facial para controle de presença automatizado de alunos, para tanto, foram necessárias muitas pesquisas tecnológicas com intuito de somar e aperfeiçoar os conhecimentos adquiridos.

A proposta desse sistema é trazer os responsáveis mais próximos da vida escolar dos alunos, de forma que estejam sempre informados em tempo real de que seus filhos estão realmente na aula, considerando que assim o sistema proporciona segurança e comodidade para os mesmos.

O sistema permite o armazenamento das informações na base de dados, de forma organizada, que poderão ser consultadas e analisadas com eficiência.

Embora o projeto ainda não tenha sido implantado nas escolas, existe a convicção de que com controles individuais automatizados pelo sistema, onde o professor tem mais tempo para lecionar, e os responsáveis tem plena ciência de onde seus filhos estão, sem dúvida isto agregará qualidade e valor a IES, ao ensino e aos alunos.

Foram realizados alguns testes individuais da ferramenta e um teste maior em um evento intitulado Fatec Portas Abertas realizado no dia 04 de maio de 2018 em uma IES na cidade de Mogi Mirim-SP. Estiveram presentes 425 pessoas entre elas 312 alunos regulares da instituição. O algoritmo implementado mostrou-se 85% de precisão reconhecendo a maioria dos alunos presentes e dispensando a tradicional chamada feita pelos professores.

Sendo assim, podemos considerar como muito bom os resultados

obtidos e a plena confirmação da hipótese levantada na introdução o qual nos leva a possibilidade de tornar a ferramenta desenvolvida como um produto a ser possivelmente comercializado. Para trabalhos futuros percebeu-se a necessidade da integração da ferramenta com as redes sociais permitindo assim ir além do e-mail e disparar “*posts*” de notificação de presença para aplicativos como *WhatsApp*, *Twitter* ou *Facebook*.

## REFERÊNCIAS

- BLANC-TALON, J. *et al.* Advanced Concepts for Intelligent Vision Systems: **18th International Conference**. Bélgica, 2017.
- BRUNELLI, R.; POGGIO, T. **Face Recognition through Geometrical Features**. European Conference on Computer Vision (ECCV), 1992, 792–800 p.
- COELHO, F. C. **Computação Científica com Python**. Petrópolis: Edição do Autor, 2007.
- DLIB C++ Library. **Face Recognition**. Disponível em < [http://dlib.net/dnn\\_face\\_recognition\\_ex.cpp.html](http://dlib.net/dnn_face_recognition_ex.cpp.html) > Acesso em abril de 2018.
- ESQUEF, I. A. **Técnicas de Entropia em Processamento de Imagens**. Rio de Janeiro, 2002. Dissertação (Mestrado em Instrumentação Científica). Centro Brasileiro de Pesquisas Físicas. Rio de Janeiro, 2002.
- GOMES, J.; VELHO, L. **Computação Gráfica: imagem**. 2ª ed. Rio de Janeiro: IMPA, 2002.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. 3ª ed. São Paulo: Editora Pearson, 2009.
- KANADE, T. **Picture processing system by computer complex and recognition of human faces**. PhD thesis, Kyoto University, 1973.
- LABAKI, J. **Introdução a Python – Módulo A**. Disponível em <http://wiki.python.org.br/DocumentacaoPython?action=AttachFile&do=get&target=material2.pdf> > Acesso em Setembro de 2017.
- MACHADO, J. L. A. **Perde-se 1/5 de uma hora aula, em média, no Brasil... O que fazer?**, São José dos Campos, 06 jan. 2016. Disponível em < <http://www.planetaeducacao.com.br/portal/artigo.asp?artigo=2686> > Acesso em maio de 2018.
- GEITGEY, A. **Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning**. 24 jul. 2016. Disponível em < <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78> > Acesso em maio de 2018.

MENEZES, N. N. C. **Introdução à programação com Python**: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec Editora, 2010

OPENCV. **Face Recognition with OpenCV**. Disponível em: < [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html) > Acesso em outubro de 2017.

SIQUEIRA, M. L. de. **Reconhecimento Automático de Padrões em Imagens Ecocardiográficas**. Tese (Doutorado em Ciência da Computação). Universidade Federal do Rio Grande do Sul. Porto Alegre, 2010.

TURATI, C. *et al.* **Newborns face recognition**: Role of inner and outer facial features. *Child Development*, 2006.

TURK, M.; PENTLAND, A. Eigen faces for recognition. **Journal of Cognitive Neuroscience**. 1991, 71–86 p.

VANDERPLAS, J. What Is Machine Learning?. *In*: VANDERPLAS, Jake. **Python Data Science Handbook**. Sebastopol, United States: O'Reilly Media, 2016. *E-book*.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. *In* **CVPR 2001 – IEEE Conference on Computer Vision and Pattern Recognition**, 2001.

RAMOS, V. **Porquê aprender Python: 5 motivos para começar agora!** Disponível em < <https://pythonacademy.com.br/blog/porque-aprender-python-5-motivos-para-aprender-agora> > Acesso em Março de 2019.

KAZEMI, V.; SULLIVAN, J. One Millisecond Face Alignment with an Ensemble of Regression Trees. **Conference: Computer Vision and Pattern Recognition, At Columbus, Ohio, USA, 2014**. Disponível em: junho 2014. Acesso em: Abril de 2018.