http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2025.v17.e432
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 26/12/20243 Revisado: 27/08/2024 Aceito: 11/02/2025

# REDE NEURAL PROFUNDA DE MÚLTIPLAS ENTRADAS COM DADOS 3D APLICADOS AO RECONHECIMENTO FACIAL

## Deep Neural Network of multiple input with 3D data applied to Face Recognition

**Luís Henrique Picinin Jandre[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Mário Augusto Pazoti[1], Almir Olivette Artero[2]**

**[1]Universidade do Oeste Paulista – UNOESTE, Faculdade de Informática de Presidente Prudente – FIPP, Presidente Prudente, SP. [2]Universidade Estadual Paulista - UNESP, Departamento de Matemática e Computação, Presidente Prudente.**
**Email:** luishpj@outlook.com.br, chico@unoeste.br, llalmeida@unoeste.br, mario@unoeste.br, almir.artero@unesp.br

**RESUMO –** O reconhecimento facial está cada vez mais sendo usado em aplicações comerciais em diversos cenários, e usa a biometria 2D ou 3D do rosto para identificar as pessoas. Neste trabalho é proposta uma metodologia utilizando rede neural multientrada para realizar o reconhecimento facial de indivíduos a partir das características 3D extraídas de imagens frontais. Para a extração das características 3D as imagens foram inicialmente submetidas à rede pix2vertex para realizar a reconstrução 3D da geometria facial de cada indivíduo. Após a reconstrução 3D foram extraídos 275 pontos, contendo as seguintes informações: coordenadas *x*, *y* e *z*, e os descritores de pontos chave do algoritmo SIFT (*Scale Invariant Feature Transform*). E por fim, essas informações são processadas em uma rede neural artificial de múltiplas entradas para a previsão da classificação de cada indivíduo. A avaliação dos resultados mostra que a rede foi capaz de classificar corretamente os indivíduos com uma precisão de 95,79% no conjunto de validação.
**Palavras-chave:** Reconhecimento facial; pix2vertex; SIFT.

**ABSTRACT –** Facial recognition is increasingly being used in commercial applications in a variety of scenarios, and uses 2D or 3D biometrics of the face to identify people. We propose a methodology using multi input neural network to perform facial recognition of the individuals from the 3D features extracted from frontal images. For 3D features extraction the images were at first submitted to the pix2vertex network in order to perform 3D reconstruction of the facial geometry of each individual. After the 3D reconstruction 275 points were extracted, to each point there are x, y and z coordinates, and the SIFT descriptor. Finally, this information is processed in a multiple-input artificial neural network to predict the classification of each individual. Evaluation of the results shows that the network was able to correctly classify individuals with an accuracy of 95.79% in the validation set.
**Keywords:** Facial recognition; pix2vertex; SIFT

## 1. INTRODUÇÃO

O reconhecimento facial é um dos campos de pesquisa mais dinâmicos da visão computacional e do reconhecimento de padrões, com muitas aplicações práticas e comerciais, incluindo identificação, controle de acesso, análise forense e interações homem-computador (Adjabi *et al*., 2020).

Além de aplicações comerciais, o reconhecimento facial pode ser usado em diversos cenários, como vigilância, marketing, contagem de números de pessoas em ambientes com classificação por características como idade ou sexo, portas interativas, entre outros. Um sistema de reconhecimento facial usa biometria para mapear recursos faciais de uma fotografia ou vídeo. As informações são comparadas com um banco de dados de rostos conhecidos para encontrar uma correspondência (Symanovich, 2023).

Embora as pesquisas de reconhecimento facial 2D tenha feito progressos significativos nos últimos anos, a precisão ainda depende muito das condições de luz e poses humana. As técnicas utilizadas ainda são desafiadas pelas limitações intrínsecas das imagens 2D, devido às variações de iluminação, pose e expressão, oclusão, disfarces, atrasos de tempo e qualidade da imagem (Jing; Lu; Gao, 2023).

A rápida evolução dos sensores 3D revela um novo caminho para o reconhecimento facial, que pode superar as limitações fundamentais das tecnologias 2D. Em comparação com as imagens 2D, as faces 3D contêm informações geométricas mais ricas, que podem fornecer mais características de discriminação e ajudar os sistemas de reconhecimento de faces a superar os defeitos e as desvantagens inerentes ao reconhecimento de faces 2D, como expressão facial, oclusão e variação de pose (Jing; Lu; Gao, 2023).

O reconhecimento de faces 3D vem incentivando cada vez mais as pesquisas em torno dos problemas envolvendo esse tipo de solução. Um exemplo disso está na indústria que está buscando mais confiabilidade no reconhecimento facial, como por exemplo, a empresa Apple com a tecnologia Face ID (APPLE, 2020), que segundo a empresa, garante autenticação intuitiva e segura, ativada pelo sistema de câmera TrueDepth de última geração que mapeia a geometria do rosto. O TrueDepth é um conjunto de sensores que são usados no iPhone e iPad Pro, capazes de criar modelos tridimensionais do rosto do usuário a partir da câmera de selfie dos dispositivos.

A alternativa de utilizar dados 3D vem sendo aplicada cada vez mais na área de reconhecimento facial, que com apenas algumas fotografias, é possível modelar a face de um indivíduo (Leal, 2017).

Segundo Adjabi *et al*., (2020), as abordagens 2D atingiram um certo grau de maturidade e registaram excelentes taxas de reconhecimento, apesar desse desempenho ser obtido em ambientes onde os parâmetros de aquisição são controlados, como a iluminação, o ângulo de visão e a distância entre a câmara e o sujeito. No entanto, se as condições ambientais (por exemplo, iluminação) ou a aparência facial (por exemplo, pose ou expressão facial) mudarem, este desempenho diminui drasticamente. As abordagens 3D foram propostas como uma solução alternativa para esses problemas, em que a vantagem reside na invariância às condições de pose e iluminação.

Este trabalho tem como contribuição utilizar técnicas de visão computacional e inteligência artificial para realizar o reconhecimento facial de indivíduos a partir de dados 3D (coordenadas *x*, *y* e *z*) e dados 2D (descritores SIFT) extraídos de imagens. Para o reconhecimento facial foi utilizada uma rede neural de múltiplas entradas, buscando o maior número de acerto possível.

Após esta seção introdutória, o trabalho está organizado da seguinte maneira. Na Seção 2 são apresentados os conceitos que fundamentam a base teórica para este trabalho. Na Seção 3 é detalhado todo o processo desenvolvido, contemplando o funcionamento do método proposto e treinamento. Na Seção 4 são apresentados a avaliação e discussão dos resultados, incluindo um experimento com a rede neural treinada. Por fim, na Seção 5 encontram-se as considerações finais.

## 2. CONCEITOS FUNDAMENTAIS

Nesta seção é apresentada a fundamentação teórica a respeito dos métodos utilizados para o desenvolvimento deste trabalho. Inicialmente, são apresentadas as técnicas para aumento de dados, aplicadas para amenizar o problema de quantidade de dados limitada, na sequência o método LBF (*Local Binary Features*) usado para definir a região de recorte na imagem, seguido pelo modelo *pix2vertex* usado para a reconstrução da malha facial 3D. Na sequência encontra-se a solução *MediaPipe Face Mesh* que realiza a detecção de pontos de referência facial, seguido do algoritmo SIFT (*Scale Invariant Feature Transform*) que foi utilizado para a obtenção dos descritores de pontos chave nas imagens 2D, e por fim, as funções de ativação *ReLU* e *Softmax* utilizadas na rede neural deste trabalho que foi construída com a biblioteca *Tensorflow*.

### 2.1. Aumento de dados

O aumento de dados é uma metodologia que permite superar a quantidade e diversidade de dados limitada, através da geração de novos dados com os originais já existentes (Tatar; Haghighi; Zeinijahromi, 2025).

No caso para imagens, esses novos dados podem ser criados por meio de diversas técnicas, sendo as mais populares *affine transformation* e modificações de cor. Com *affine transformation* tem-se as técnicas: rotação, espelhamento, escala (zoom *in*/*out*) e cisalhamento. Já como modificações de cor tem-se: alterações no contraste, equalização de histograma, equilíbrio de branco (*white balance*) e nitidez (*sharpen*) (Mikołajczyk; Grochowski, 2018).

Neste trabalho foram escolhidas as técnicas para aumento de dados de rotação e alteração de brilho de imagem.

### 2.2. LBF

O LBF (*Local Binary Features*) é um método usado para detecção de pontos chave, onde as características locais binárias são geradas a partir de uma função de mapeamento de características (Ren *et al*., 2014).

Essa função de mapeamento é composta por um conjunto de funções de mapeamento de características locais altamente discriminativas que podem ser aprendidas de forma independente por meio de treinamento com regressões lineares (Ren *et al*., 2014).

A quantidade de pontos detectados é determinada pelo número de funções de mapeamento de características locais, onde cada ponto possui função de mapeamento local.

O conjunto de funções de mapeamento de características locais pode ser descrito como $\varphi^t = [\varphi_1^t, \varphi_2^t, \ldots, \varphi_L^t]$ onde *L* é o número de pontos.

A matriz de regressão linear $W^t$ é produzida por meio de regressões lineares sobre as características locais binárias retornadas pelas funções de mapeamento.

O processo de treinamento (Ren *et al*., 2014) para aprender as características binárias locais é dado por uma somatória, onde *i* são as interações sobre todas as amostras de treinamento, $\pi_l$ extrai dois elementos do vetor $\Delta\hat{S}_i$ e $\pi_l \circ \Delta\hat{S}_i^t$ é a distância 2D do 1º ponto de referência na *i*-ésima amostra. Todo esse treinamento para aprender a mapear as características binárias locais $\varphi^t$ é representado pela Equação 1.

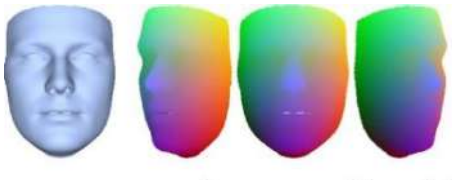
$$\min_{w^t, \phi_l^t} \sum_{i=1} \left\| \pi_l \circ \Delta\hat{S}_i^t - w_l^t \phi_l^t\left(I_i, S_i^{t-1}\right) \right\|_2^2 \qquad (1)$$

### 2.3. Pix2vertex

O pix2vertex (Sela; Richardson; Kimmel, 2017) é uma solução que permite a reconstrução irrestrita da geometria facial com a utilização de apenas uma imagem de entrada, através de tradução *image-to-image*, obtendo como saída uma representação geométrica da face composta por dois itens, sendo estes o mapa de profundidade e o mapa de correspondência. O mapa de correspondência permite mapear os pixels da imagem para pontos em um *template* de modelo facial, fornecido como uma malha triangulada.

Para possibilitar o cálculo para qualquer geometria facial é usado um modelo facial de referência normalizada apresentado na Figura 1. Cada pixel do mapa de correspondência recebe a cor de um vértice 3D do modelo facial de referência, possibilitando mapear a face na imagem de entrada para o modelo de referência por meio do mapa de correspondência.

**Figura 1.** Modelo facial de referência.



Fonte: (Sela; Richardson; Kimmel, 2017).

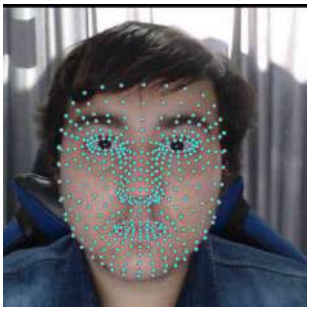
Com base na imagem de profundidade e no mapa de correspondência, obtém-se uma representação, que através de deformação iterativo-elástico seguido por um refinamento de detalhes, é convertida em 3D na forma de uma malha facial precisa (Sela; Richardson; Kimmel, 2017).

Essa técnica é útil para vários aplicativos de processamento facial dinâmico, como facial *rigs* (Danieau *et al.*, 2019), que são faces manipuláveis ou animadas, que permitem a criação e edição de animações fotorealistas (Sela; Richardson; Kimmel, 2017).

### 2.3. MediaPipe Face Mesh

A solução *MediaPipe Face Mesh* (Mediapipe, 2022) possibilita, com apenas uma câmera, a estimativa de 468 pontos de referência facial 3D mostrados na Figura 2. Essa solução é otimizada e possibilita execução em tempo real até mesmo em aparelhos móveis, sem a necessidade de utilizar qualquer tipo de sensor de profundidade, como por exemplo, câmeras ToF (*Time of Flight*) (Wübbenhorst; Wermke; Meffert, 2020), LiDAR (*Light Detection And Ranging*) (Zalevsky *et al.*, 2021), Kinect (Zhang, 2012), entre outros.

**Figura 2.** Exemplo de estimativa de 468 pontos de referência facial 3D.



Fonte: Os autores.

*Face Mesh* possui foco principalmente voltado para aplicações envolvendo realidade aumentada, que faz a predição do rosto no espaço 3D através de estimativa usando os pontos de referência do rosto e suas coordenadas na tela.

Os dados geométricos da face consistem em primitivas 3D comuns, como matriz de transformação de pose de face e malha de face triangular.

Após a predição, é feito um refinamento para tratar o problema de registro rígido entre diferentes formas de entrada, representadas por correspondências de pontos, conhecido na estatística e literatura de análise de forma como Análise de Procrustes (Pizarro; Bartoli, 2011), a fim de tornar a solução mais robusta (Mediapipe, 2022).

### 2.4. SIFT

O algoritmo SIFT (*Scale Invariant Feature Transform*) é um método muito eficiente para identificar e descrever pontos chave em imagens, resultando em um vetor com 128 valores, descrevendo cada ponto chave da imagem. Adicionalmente, o algoritmo SIFT fornece a posição *x*, *y*, escala *s* e orientação $\theta$ para todos os pontos chave (Lowe, 2004).

O algoritmo pode ser dividido em duas partes, o detector e o descritor. O detector baseia-se em cálculos de diferença de filtros Gaussianos, enquanto o descritor utiliza-se de histogramas de gradientes orientados para descrever a vizinhança local dos pontos de interesse. Seu funcionamento pode ser subdividido em quatro etapas principais, sendo que as duas primeiras se referem a parte do detector dos pontos chave, e as duas últimas ao descritor.

A primeira etapa, detecção de extremos no espaço escala, busca pontos que são invariáveis à escala e orientação ao se utilizar de diferença de filtros Gaussianos. O espaço escala de uma imagem pode ser definido pela Equação 2, onde $G(x, y, \sigma)$ representa uma convolução de uma Gaussiana de escala variável e $I(x, y)$ representa a imagem de entrada (Koenderink, 1984; Lindeberg, 1994). A gaussiana $G(x, y, \sigma)$ pode ser definida pela Equação 3.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3)$$

Para a detecção de pontos chave estáveis no espaço escala, em (Lowe, 1999) é sugerido o uso de extremos do espaço escala com o uso de uma função de diferença de gaussianas (DoG) (Lowe, 1999), que é calculada pela Equação 4, onde *k* é uma constante multiplicadora (Mikołajczyk; Schmid, 2005).

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (4)$$

Na segunda etapa, localização de pontos chave, a partir de *D*(*x*, *y*, *σ*), o algoritmo percorre áreas nas quais foram detectados extremos (máximos e mínimos), e detecta pontos chave comparando cada pixel com seus oito vizinhos da escala atual e nove vizinhos das escalas superior e inferior, totalizando 26 vizinhos. O SIFT garante que os pontos chave são localizados em regiões e escalas com alta variação, o que os torna estáveis para caracterizar a imagem (Lowe, 2004).

A etapa que determina a orientação utiliza o gradiente local da imagem para determinar uma ou mais orientações para cada ponto chave, onde são utilizados para prover a invariância em orientação, escala e localização. A magnitude do gradiente pode ser calculada pela Equação 5.

$$m(x, y) = \sqrt{\Delta x^2 + \Delta y^2} \quad (5)$$

onde $\Delta x = L(x + 1, y) - L(x - 1, y)$ e $\Delta y = L(x, y + 1) - L(x, y - 1)$. A orientação $\theta(x,y)$ é calculada pela Equação 6.

$$\theta(x, y) = \arctan(\frac{\Delta y}{\Delta x}) \quad (6)$$

A última etapa do algoritmo é a descrição dos pontos chave. Nessa etapa é calculado um descritor para cada região da imagem local, que é distinta e invariante a variações adicionais, tais como mudanças na iluminação ou ponto de vista 3D. Na Figura 3 é apresentado um exemplo de imagem contendo os pontos chave detectados e descritos pelo algoritmo SIFT.

**Figura 3.** Pontos chave detectados e descritos pelo algoritmo SIFT.

Fonte: Os autores.

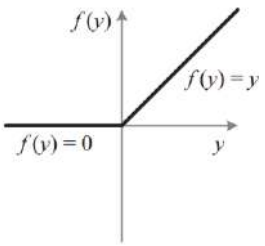
## 2.5. ReLU

ReLU (*Rectified Linear Unit*) é uma função de ativação usada em redes neurais utilizada em diversos trabalhos (Araújo *et al.*, 2017; Szegedy *et al.*, 2017), e pode possibilitar um maior número de camadas ocultas, ao contrário da função logística (Hu *et al.*, 2019).

A definição da função de ativação ReLU é dada pela Equação 7 e representada pela Figura 4.

$$f(x) = max(0, x) \quad (7)$$

**Figura 4.** Função de ativação ReLU (HE *et al*., 2015).




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

### 2.6. Softmax

A função *Softmax* (Chollet *et al.*, 2023) realiza a conversão de uma lista de valores (neurônios da camada) em uma distribuição de probabilidade. Cada elemento da lista é convertido no intervalo de zero a um. A função é representada pela Equação 8.

$$softmax(x_i) = \frac{exp(x_i)}{\sum_{k=1}^{n} exp(x_k)} \quad (8)$$

Essa conversão pode ser feita em cada um dos elementos da lista, calculando-se o expoente de cada elemento dividido pela soma dos expoentes de todos os elementos (Jones *et al.*, 2001). Ao final é obtida uma probabilidade para cada índice da lista.

## 3. MÉTODO PROPOSTO

Nesta seção é descrito o funcionamento do método proposto, sendo dividido em quatro etapas. As etapas são: criação do *dataset* de imagens utilizado, funcionamento do método proposto, treinamento e avaliação dos resultados, e um experimento realizado com os pesos da rede já treinada.

### 3.1. *Dataset* de imagens

O *dataset* utilizado neste trabalho foi produzido a partir do banco de imagens FEI Face Database (Thomaz, 2006). Um banco de dados facial brasileiro constituído de 14 imagens para cada um dos 200 indivíduos existentes. Na Figura 5 é mostrado um exemplo de imagens de um indivíduo. As imagens possuem resolução de 640x480 pixels e foram capturadas a partir de uma câmera em posição fixa. Esse banco de imagens está disponível em Thomaz (2006).

Inicialmente, antes da montagem do *dataset* usado no treinamento da rede, o banco de imagens original foi separado em dois grupos, sendo um para o treinamento e validação com 198 indivíduos e o outro para o experimento da rede com dois indivíduos.

**Figura 5.** Exemplo de imagens de um indivíduo do FEI Face Database (Thomaz, 2006), demonstrando as diferentes variações do mesmo.

Fonte: (Thomaz, 2006).

Na montagem do *dataset* usado neste trabalho foram selecionadas quatro imagens frontais de cada um dos 198 indivíduos do FEI Face Database (Thomaz, 2006). A Figura 6 contém um exemplo das quatro imagens selecionadas, e dentre essas imagens, uma foi separada aleatoriamente para compor conjunto de validação da rede.

**Figura 6.** Exemplo das quatro imagens frontais selecionadas de um indivíduo do FEI Face Database.

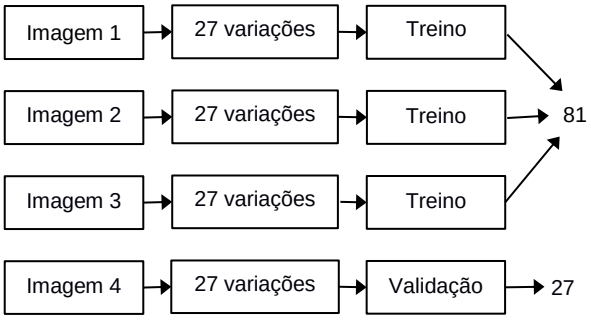
Fonte: (Thomaz, 2006).

Após essa separação, cada uma das quatro imagens foi submetida separadamente ao aumento de dados, originando 81 imagens para treinamento e 27 para validação.

O aumento de dados se dá na seguinte forma, para cada uma das quatro imagens originais obtém-se 27 imagens (original mais 26 variações) através da seguinte ordem de operações. A partir de uma imagem, produz-se oito novas imagens com rotações (total de nove considerando a original). Em seguida, a partir de cada uma das nove imagens tem-se a geração de duas novas imagens, com brilho aumentado e

reduzido (total de três imagens considerando a imagem original), gerando assim 27 imagens. Na Figura 7 é mostrado o processo simplificado, contendo a distribuição dos dados entre os conjuntos para treinamento e validação.

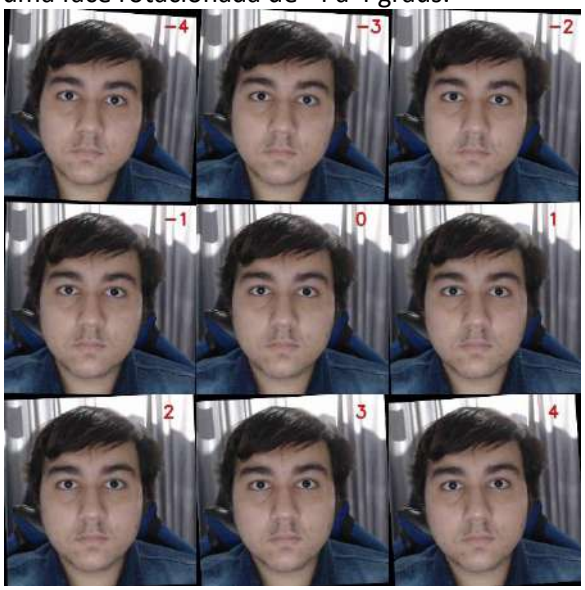
**Figura 7.** Representação do aumento de dados e separação dos conjuntos de treinamento e validação.




Fonte: Os autores.

As técnicas de aumento de dados usadas para obter variações das quatro imagens originais foram: rotação de -4 a 4 graus (de 1 em 1 grau) e redução e aumento de brilho. Na Figura 8 são mostradas as imagens obtidas a partir das rotações, contendo o ângulo utilizado na rotação escrito na parte superior direita em vermelho de cada imagem.

**Figura 8.** Exemplo de imagens de uma face rotacionada de -4 a 4 graus.

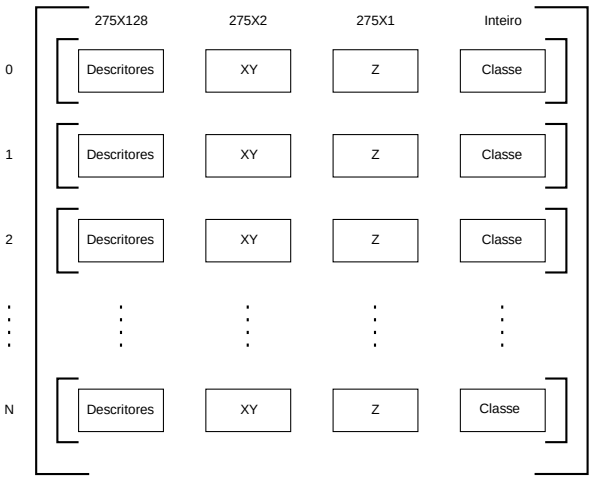



Fonte: Os autores.

Para cada umas das imagens geradas a partir do aumento de dados, foram obtidas as seguintes informações: coordenadas *x*, *y* e *z*, descritores SIFT e a classe à qual a imagem pertence (representando cada indivíduo). Essas informações são armazenadas durante o processo de extração separadamente em duas listas. Na primeira lista foram armazenadas as informações contidas nas 81 imagens de cada indivíduo (três imagens originais e imagens geradas com o aumento de dados) selecionadas para treinamento. Na segunda lista foram armazenadas as informações contidas nas 27 imagens selecionadas para validação (uma imagem original e imagens geradas com o aumento de dados).

Na sequência, as listas foram armazenadas separadamente em dois arquivos NPY (NumPy). O propósito de armazenar as informações extraídas em arquivos (“descritores”, “*xy*”, “*z*” e “classe”) é permitir reconstruir as listas, posteriormente, contendo os dados já processados e separados nos conjuntos de treinamento e validação em qualquer computador. Dessa forma, evita ter que refazer o processamento do conjunto inteiro toda vez que o modelo proposto tiver que ser submetido à fase de treinamento e validação. A estrutura dos dois arquivos é apresentada na Figura 9.

**Figura 9.** Representação da estrutura do arquivo utilizado para armazenar os dados (“descritores”, “*xy*”, “*z*” e “classe”) das imagens pertencente a cada indivíduo do treinamento e validação da rede neural de múltipla entrada.

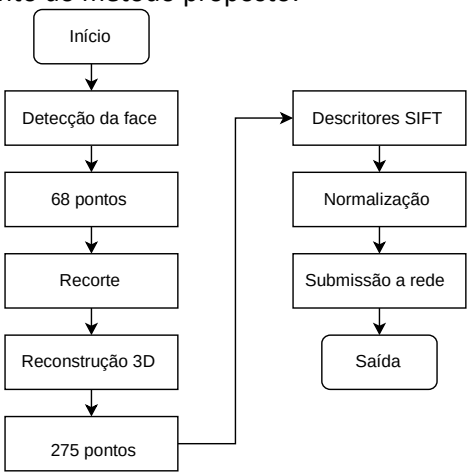



Fonte: Os autores.

### 3.1. Funcionamento do método proposto

O método proposto foi dividido em oito etapas principais, sendo elas a detecção do rosto, detecção de 68 pontos de referência com o LBF (Ren *et al.*, 2014), recorte da face, reconstrução 3D com o uso do pix2vertex (Sela; Richardson; Kimmel, 2017), detecção dos 275 pontos de referência da imagem recortada com o FaceMesh (Mediapipe, 2022), extração dos descritores SIFT (Lowe, 2004) dos 275 pontos de referência, normalização dos dados e submissão à rede de múltipla entrada. O fluxograma de funcionamento do método proposto é mostrado na Figura 10.

**Figura 10.** Fluxograma de funcionamento do método proposto.

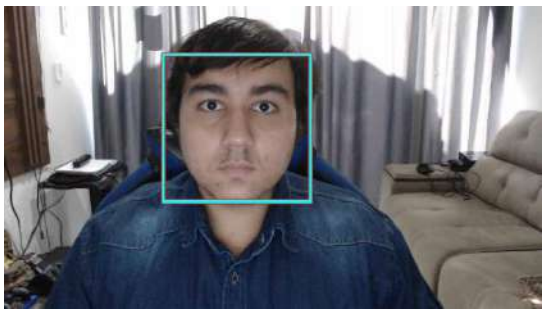



Fonte: Os autores.

O método proposto tem como objetivo realizar a classificação correta dos indivíduos com o uso das seguintes informações: descritores SIFT e coordenadas *x*, *y* e *z*.

Inicialmente, foi realizada a leitura das imagens para a detecção das faces com uso do classificador em cascata proposto por Viola e Jones (Viola; Jones, 2001), como mostrado na Figura 11.

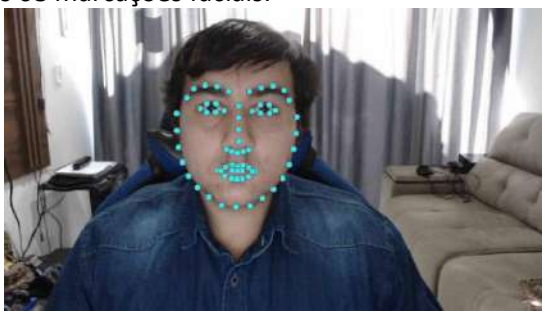
**Figura 11.** Exemplo de face detectada com o uso do método de Viola e Jones (VIOLA; JONES, 2001).



Fonte: Os autores.

Na sequência, a partir da região demarcada pelo retângulo entorno da face detectada, foi utilizado o algoritmo LBF (Ren *et al*., 2014) para detecção de 68 pontos (marcações faciais). Quatro desses pontos, os localizados nas extremidades (esquerda, direita, cima e baixo), foram usados para a fase de recorte da face. Os 68 pontos são mostrados na Figura 12.

**Figura 12.** Exemplo de detecção das 68 marcações faciais.



Fonte: Os autores.

Neste trabalho foi utilizado um modelo LBF (Ren *et al*., 2014) já treinado para a detecção de marcações faciais, e por isso os quatro pontos externos são localizados dentre os 68 para a definição da região mínima, esse modelo já treinado está disponível em (Kurnianggoro; Passalacqua, 2020).

Com base nos 68 pontos detectados (Figura 12), os quatro pontos mais externos foram usados para definir a região de interesse na forma de um novo retângulo. Para esse novo retângulo têm-se o *x* inicial definido pela Equação 9, *y* inicial é definido pela Equação 10 e altura e largura sendo definidos pelas Equações 11 e 12.

$$x = center_x - \max(largura, altura)\ /2 \quad (9)$$

$$y = min_y - altura * 0.3 \quad (10)$$

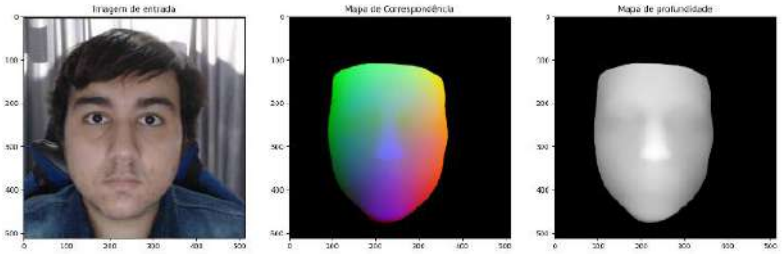
$$altura = (max_y - min_y) * 1.2 \quad (11)$$

$$largura = (max_x - min_x) * 1.2 \quad (12)$$

Após o retângulo ter sido definido é calculada a diferença entre altura e largura, em seguida a menor dimensão é completada com um preenchimento para que a altura e largura sejam iguais, formando assim um novo quadrado. Em seguida o novo quadrado é redimensionado para dimensão de 512x512 pixels, e a imagem é recortada com essa dimensão, que é a dimensão aceita na entrada da rede pix2vertex (Sela; Richardson; Kimmel, 2017).

Após o redimensionamento e recorte, a imagem é submetida a rede neural pix2vertex (Sela; Richardson; Kimmel, 2017), retornando assim um mapa correspondência e um mapa de profundidade para a reconstrução 3D. A imagem de entrada e as imagens de saída são representadas na Figura 13 e a reconstrução 3D na Figura 14.

**Figura 13.** Exemplo de entrada, mapa de correspondência e profundidade.



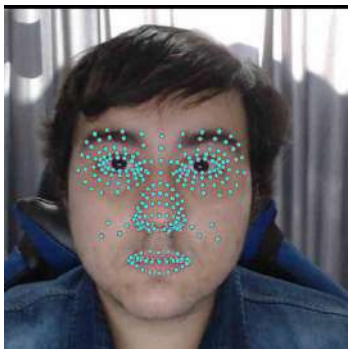

Fonte: Os autores.

**Figura 14.** Exemplo de reconstrução 3D.

Fonte: Os autores.

Após a execução do pix2vertex, a imagem recortada é processada pelo Face Mesh (Mediapipe, 2022), retornando assim 468 pontos de interesse (Figura 2). São selecionados apenas 275 pontos mais relevantes, contidos dentro da região de interesse, a fim de reduzir o número de dados que serão necessários na entrada da rede. Esses pontos são representados na Figura 15.

**Figura 15.** Exemplo dos 275 pontos selecionados.



Fonte: Os autores.

Os 275 pontos foram escolhidos visando as regiões que contêm características que fazem parte da composição de uma face, sendo elas olhos, nariz, boca e alguns pontos espalhados entre o nariz e as orelhas. Essas características são as mais discrepantes entre os indivíduos.

A partir dos 275 pontos selecionados, foi utilizada apenas a segunda parte do algoritmo SIFT (descrição dos pontos chave detectados), para a extração dos descritores de cada ponto.

A normalização dos descritores, coordenadas *x*, *y* e *z* fez-se necessária, uma vez que cada uma das três entradas da rede teria faixas de valores diferentes, o que poderia impactar de forma negativa no resultado do modelo. Outro motivo é que além de reduzir os erros de modelagem, a normalização acelera o processo de treinamento da rede neural e leva à simplificação do sistema como um todo (Koval, 2018). A normalização é descrita pela Equação 13.

$$normalized = (value - min)/(\max - min) \qquad (13)$$

Para a classificação dos indivíduos, foi utilizada uma rede neural artificial de múltiplas entradas (Livieris; Dafnis; Kalivas, 2020). Esse tipo de rede foi utilizado em outros trabalhos (Bhardwaj; Gohel; Namuduri, 2019; Raza *et al.*, 2017). A rede neural do modelo proposto neste trabalho está representada da na Figura 16, onde cada entrada possui as seguintes informações: “descritores” de ponto chave, coordenadas “*xy*” e “*z*”.

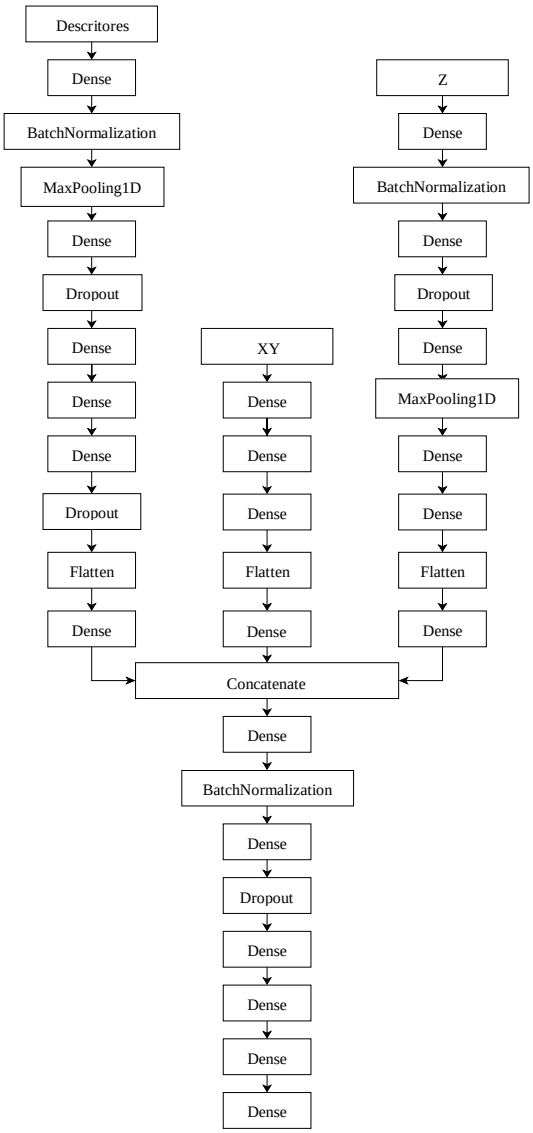
Modelar a rede neural dessa forma possibilita atribuir quantidades de neurônios e camadas diferentes de acordo com cada entrada, e por isso, esse tipo de rede foi escolhida.

Cada uma das três entradas passa pelo processo de treinamento separada uma das outras na forma de três sub-redes (“descritores”, “*xy*” e “*z*”) e posteriormente tem-se a última camada de todas as três concatenadas em uma camada densa.

No caso do modelo proposto, a quantidade de neurônios da última camada de cada sub-rede possui uma quantidade de neurônios diferentes, na seguinte ordem:

1. Descritores com 2000 neurônios;
2. XY com 500 neurônios;
3. Z com 1000 neurônios.

**Figura 16.** Representação em forma de diagrama do modelo proposto da rede neural de múltipla entrada.




Fonte: Os autores.

Na concatenação (Chollet, *et al*., 2023) é gerada com as últimas camadas das três sub-redes, uma nova camada com 3500 neurônios contendo os pesos das últimas três camadas anteriores à concatenação, como mostrado na Figura 16.

A última camada do modelo proposto possui a função *softmax* com 198 neurônios. Dessa forma, tem-se como saída a probabilidade de reconhecimento dos 198 indivíduos submetidos ao processo de treinamento da rede.

### 3.2. Treinamento e avaliação dos resultados

Para o treinamento da rede neural, foi realizada a reconstrução das duas listas armazenadas na montagem do *dataset*, a partir dos arquivos NPY (NumPy). Uma das listas contêm 16.038 (81 imagens * 198 indivíduos) entradas para o treinamento da rede e a na outra lista contêm 5.346 (27 imagens * 198 indivíduos) entradas para validação da rede, sendo que para cada entrada da rede tem-se "descritores", "*xy*" e "*z*".

Além disso, alguns parâmetros fixos foram necessários para realizar o treinamento, esses parâmetros são chamados de hiper parâmetros, pois não mudam, porém, são essenciais para a eficiência da rede neural. Os hiper parâmetros usados foram: taxa de aprendizado de 3e-4; otimizador Adam; tamanho do lote definido como 32; número de épocas igual a 20. Sendo a taxa de aprendizado sugerida por Karpathy (2019) e o restante definido empiricamente, a partir de testes realizados durante a modelagem e treinamento da rede neural.

Após a definição dos hiper parâmetros a rede neural foi submetida ao processo de treinamento dentro de um *loop* e avaliado ao final para coleta dos resultados.

Em cada *loop*, a rede é treinada com as informações separadas para treinamento na montagem do *dataset* e ao final do *loop* a rede é avaliada com as informações de validação também separados na montagem do *dataset*.

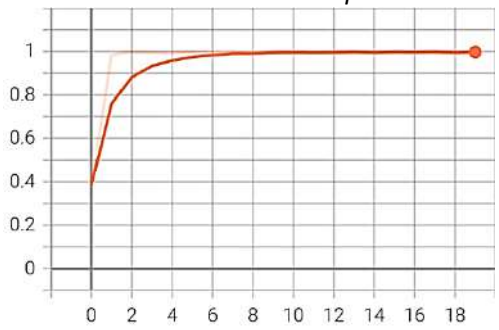
Caso os resultados coletados da validação sejam insatisfatórios, a rede então é novamente instanciada e tem seus pesos reinicializados aleatoriamente voltando para o treinamento não mantendo nenhum dado do *loop* anterior.

No treinamento do modelo de rede neural proposto foi utilizado um computador com 8 GB de memória RAM, CPU i5-7400 3GHz e GPU GTX 1060 com 6 GB de memória. O tempo médio da execução de um *loop* (apenas as 20 épocas) foi de 23 minutos e 54 segundos e o tempo médio de execução de uma predição foi de 94,15 milissegundos.

## 4. RESULTADOS E DISCUSSÃO

O treinamento da rede neural foi finalizado após 20 épocas, onde foi percebido que a acurácia praticamente se estabilizou em um patamar satisfatório. A acurácia da rede ao final de cada uma das 20 épocas de um *loop* é representada pela linha mais clara e a suavização em cor escura na Figura 17.

**Figura 17.** Representação do gráfico de acurácia no último *loop* de treinamento.




Fonte: Os autores.

Para a avaliação dos resultados foi levado em consideração a acurácia e o desempenho do modelo proposto neste trabalho.

A acurácia foi verificada com base na taxa de acerto do modelo proposto, classificando corretamente o conjunto de dados separados para validação na montagem do *dataset*. Já o desempenho foi com base no tempo de execução que o modelo levou para retornar as predições.

O método proposto atingiu um resultado satisfatório no quinto *loop*, sendo possível classificar corretamente 95,79% do conjunto de validação. O resultado da validação pode ser visualizado na Tabela 1.

**Tabela 1.** Resultado da validação.

| Total de casos | Erros | Taxa de acerto | Taxa de erro |
|---|---|---|---|
| 5346 | 225 | 95.79% | 4.21% |

Fonte: Os autores.

Além do experimento de reconhecimento de faces, também foi realizado outro experimento buscando avaliar os casos de falsos positivos, por meio da predição de imagens de dois indivíduos que não foram utilizados nas fases de treinamento e validação da rede, de forma que fosse possível simular o estado da mesma ao tentar reconhecer indivíduos desconhecidos no mundo real.

A análise do experimento foi realizada com base no percentual de certeza que a rede previu erroneamente para cada submissão. Os resultados obtidos informando a probabilidade de certeza, quantidade de vezes que ocorreu, bem como o percentual de ocorrência pode ser visto na Tabela 2.

**Tabela 2.** Resultado do experimento.

| Certeza | Ocorrências | Percentual |
|---|---|---|
| **=100%** | 0 | 0% |
| **>99%** | 4 | 1.99% |
| **>98%** | 11 | 5.47% |
| **>97%** | 22 | 10.95% |
| **>96%** | 31 | 15.42% |
| **>95%** | 43 | 21.39% |
| **>94%** | 51 | 25.37% |
| **>93%** | 58 | 28.86% |
| **>92%** | 63 | 31.34% |
| **>91%** | 70 | 34.83% |
| **>90%** | 75 | 37.31% |

Fonte: Os autores.

Foi feita uma comparação da abordagem deste artigo com outros trabalhos que fazem uso do mesmo banco de imagens FEI Face Database (Thomaz, 2006).

No trabalho de Samet, Shokouh e Baskurt (2016), o sistema proposto faz uso de 2DPCA (2D *Principal Component Analysis*) para extrair características e reduzir a dimensionalidade, seguindo do uso de técnicas de regressão para gerar o vetor de características da face frontal a partir das imagens faciais, e por fim, é usada a técnica de distância de Mahalanobis para reconhecimento. Foram selecionadas imagens de 80 indivíduos, quatro imagens com poses diferentes, totalizando 320 imagens com ângulos de variação de pose de +22,5, –22,5, +45 e –45. Os resultados de acurácia obtidos foram respectivamente 87%, 90%, 69% e 75% para cada variação de pose.

Já em um trabalho mais recente, o de Masud *et al*. (2020) que utiliza modelos de *deep learning* baseados em árvores para reconhecimento facial, os resultados dos experimentos demonstram precisão de 98,65%, porém com as imagens originais do *dataset* que são 2D.

## 5. CONSIDERAÇÕES FINAIS

Os resultados obtidos com este trabalho demonstram que o método proposto pode ser aplicado para predição de faces de indivíduos e classificá-los com uma acurácia satisfatória de 95,79%.

Embora o tempo necessário para a predição seja considerado satisfatório, o sistema mostra seu gargalo na fase de reconstrução 3D. Porém, vale ressaltar que em casos de sistemas de segurança, soluções com maior precisão e menor desempenho podem prover um maior grau de segurança.

Como trabalhos futuros, possíveis melhorias podem ser realizadas com a montagem de um *dataset* com maior resolução, bem como melhorias de desempenho relacionado ao tempo de execução com futuras implementações ou técnicas para a fase de reconstrução 3D.

## REFERÊNCIAS


ADJABI, I.; OUAHABI, A.; BENZAOUI, A.; TALEB-AHMED, A. Past, Present, and Future of Face Recognition: A Review. **Electronics**, v. 9, n. 8, 2020. DOI: https://doi.org/10.3390/electronics9081188.

APPLE**. Sobre a tecnologia avançada do Face ID: Saiba** como o Face ID ajuda a proteger as informações no iPhone e no iPad Pro. Março de 2020. Disponível em: https://support.apple.com/pt-br/HT208108. Acesso em: 2 nov. 2023.

ARAÚJO, F. H. D. *et al*. Redes neurais convolucionais com tensorflow: teoria e pratica. *In*: ESCOLA REGIONAL DE INFORMÁTICA DO PIAUÍ. 3., **Livro Anais - Artigos e Muinicursos**, v. 1, n.. 1, p. 382-406, 2017.

BHARDWAJ, S.; GOHEL, H.; NAMUDURI, S. A Multiple-Input Deep Neural Network Architecture for Solution of One-Dimensional Poisson Equation. **IEEE Antennas and Wireless Propagation Letters**, v. 18, n. 11, p. 2244-2248, Nov. 2019. DOI: https://doi.org/10.1109/LAWP.2019.2933181

CHOLLET, F. *et al*. **Keras**, 2023. Disponível em: https://keras.io/. Acesso em: 26 nov. 2023.

DANIEAU, F. *et al*. Automatic Generation and Stylization of 3D Facial Rigs. *In*: IEEE CONFERENCE ON VIRTUAL REALITY AND 3D USER INTERFACES (VR), 2019, Osaka, Japan. **Anais [...].** Osaka, Japan, 2019. p. 784-792. DOI: https://doi.org/10.1109/VR.2019.8798208

SILVA, F. A.; PEREIRA, D. R.; SILVA, J. F. C.; ARTERO, A. O.; PITERI, M. A. TSRS - A new approach for traffic sign recognition using the SIFT algorithm. **Journal of Urban and Environmental Engineering**, v. 2, n. 2, p. 1-5, 2017.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2015, Santiago, Chile. **Anais [...]** Santiago, Chile, 2015. p. 1026-1034. DOI: https://doi.org/10.1109/ICCV.2015.123

HU, X.; NIU, P.; WANG, J.; ZHANG, X. A Dynamic Rectified Linear Activation Units. **IEEE Access**, v. 7, p. 180409-180416, 2019. DOI: https://doi.org/10.1109/ACCESS.2019.2959036

JAIN, A. Understanding Convolutional Neural Networks (CNNs) with an Example on the MNIST Dataset. Disponível em: https://medium.com/@abhishekjainindore24/understanding-convolutional-neural-networks-cnns-with-an-example-on-the-mnist-dataset-a64815843685. Acesso em: 28 set. 2024.

JING, Y.; LU, X.; GAO, S. 3D face recognition: A comprehensive survey in 2022. **Computational Visual Media**. v. 9, p. 657–685, 2023. DOI: https://doi.org/10.1007/s41095-022-0317-1.

JONES, E. *et al*. **SciPy**, 2001. Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.softmax.html. Acesso em: 15 fev. 2023.

KARPATHY, A. **A Recipe for Training Neural Networks** 2019. Disponível em: http://karpathy.github.io/2019/04/25/recipe/. Acesso em: 8 mar. 2023.

KOENDERINK, J. J. The structure of images. **Biological Cybernetics**, v. 50, p. 363-396, 1984. DOI: https://doi.org/10.1007/BF00336961

KOVAL, S. I. Data preparation for neural network data analysis. *In*: IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018, Moscow and St. Petersburg, Russia.

**Anais [...]** Moscow and St. Petersburg, Russia, 2018. p. 898-901. DOI: https://doi.org/10.1109/EIConRus.2018.8317233.

KURNIANGGORO, L.; PASSALACQUA, D. **Facemark API for OpenCV**, 15 jun. 2020. Disponível em: https://github.com/kurnianggoro/GSOC2017. Acesso em: 1 abr. 2022.

LEAL, J. F. S. Criação de Modelos Faciais 3D Através de Fotografias. *In*: INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO, 2017. **Livro Anais** – Artigos e Mini cursos, v. 1, n. 1, p. 5-17.

LINDEBERG, T. Scale-space theory: a basic tool for analyzing structures at different scales, **Journal of applied Statistics**, v. 21, n. 1-2, p. 225-270, 1994. DOI: https://doi.org/10.1080/757582976.

LIVIERIS, I. E.; DAFNIS, S. D.; P., G. K.; KALIVAS, D. P. A Multiple-Input Neural Network Model for Predicting Cotton Production Quantity: A Case Study. **Algorithms**, v. 13, n. 11, 2020. DOI: https://doi.org/10.3390/a13110273

LOWE, D. G. Object recognition from local scale-invariant features. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, Kerkyra,1999, Greece. **Anais** […].Kerkyra, Greece, 1999. p. 1150-1157. DOI: https://doi.org/10.1109/ICCV.1999.790410

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n. 2, p. 91-110, 2004. DOI: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

MASUD, M.; MUHAMMAD, G.; ALHUMYANI, H.; ALSHAMRANI, S. S.; CHEIKHROUHOU, O.; IBRAHIM, S.; HOSSAIN, M. S. Deep learning-based intelligent face recognition in IoT-cloud environment. **Computer Communications**, n. 152, p. 215–222, 2020. DOI: https://doi.org/10.1016/j.comcom.2020.01.050.

MEDIAPIPE. **FaceMesh**. Disponível em: https://google.github.io/mediapipe/solutions/face_mesh.html. Acesso em: 20 jan. 2022.

MIKOŁAJCZYK, A.; GROCHOWSKI, M. Data augmentation for improving deep learning in image classification problem. *In*: INTERNATIONAL INTERDISCIPLINARY PHD WORKSHOP (IIPhDW), 2018, Poland. **Anais [...].** Poland, 2018. p. 117-122. DOI: https://doi.org/10.1109/IIPHDW.2018.8388338.

MIKOŁAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. **IEEE Transaction on Pattern Analysis and Machine Intelligence**, v. 27, n. 10, p. 1615-1630, 2005. DOI: https://doi.org/10.1109/TPAMI.2005.188.

PIZARRO, D.; BARTOLI, A. Global optimization for optimal generalized procrustes analysis. *In*: Conference on Computer Vision and Pattern Recognition (CVPR), 2011, Colorado Springs, USA. **Anais [...].** Colorado Springs, USA, 2011. p. 2409-2415. DOI: https://doi.org/10.1109/CVPR.2011.5995677.

RAZA, S. E. A. *et al*. MIMO-Net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images. *In*: IEEE 14TH INTERNATIONAL SYMPOSIUM ON BIOMEDICAL IMAGING (ISBI 2017), 2017. **Anais** […]. Melbourne, Australia: IEEE, 2017. p. 337-340.

SAMET, R.; SHOKOUH, G. S.; BATUHAN, K. B. An Efficient Pose Tolerant Face Recognition Approach. **Lecture Notes in Computer Science**. Transactions on Computer Science XXVI. p. 1-12, 2016. DOI: https://doi.org/10.1007/978-3-662-49247-5_10.

REN, S.; CAO, X.; WEI, Y.; SUN, J. Face Alignment at 3000 FPS via Regressing Local Binary Features. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2014, Columbus, USA. **Anais [...]** Columbus, USA, 2014. p. 1685-1692. DOI: https://doi.org/10.1109/CVPR.2014.218.

SELA, M.; RICHARDSON, E.; KIMMEL, R. Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2017, Venice, Italy. **Anais [...]** Venice, Italy, 2017. p. 1585-1594. DOI: https://doi.org/10.1109/ICCV.2017.175.

SYMANOVICH, S. **How does facial recognition work?** 2023. Disponível em: https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html. Acesso em: 2 nov. 2023.

SZEGEDY, C. *et al*. Inception-v4, inception-ResNet and the impact of residual connections on learning. *In:* **A**AAI CONFERENCE ON ARTIFICIAL INTELLIGENCE **(AAAI'17)**.31. AAAI Press, 2017, San Francisco, California, USA. **Proceedings [...]**. San Francisco, CA, 2017, p.4278–4284. DOI: https://doi.org/10.1609/aaai.v31i1.11231

TATAR, A.; HAGHIGHI, M.; ZEINIJAHROMI, A. Experiments on image data augmentation techniques for geological rock type classification with convolutional neural networks. **Journal of Rock Mechanics and Geotechnical Engineering**. v. 17, n. 1, p. 106-125, 2025 DOI: https://doi.org/10.1016/j.jrmge.2024.02.015.

THOMAZ, C. E. **FEI Face Database**, mar. 2006. Disponível em: https://fei.edu.br/~cet/facedatabase.html. Acesso em: 17 dez. 2022.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. *In:* IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). Kauai, USA **Proceedings** [...]. Kauai, 2001. p. I-I.

WÜBBENHORST, T.; WERMKE, F.; MEFFERT, B. Synchronization of Multiple Time-of-Flight Cameras Using Photodiodes. **IEEE SENSORS**, Rotterdam, Netherlands, 2020, p. 1-4. DOI: https://doi.org/10.1109/SENSORS47125.2020.9278774.

ZALEVSKY, Z.; BULLER, G. S.; CHEN, T.; COHEN, M.; BARTON-GRIMLEY, R. Light detection and ranging (lidar): introduction. **Journal of the Optical Society of America B.** v. 38, n. 11, 2021. DOI: https://doi.org/10.1364/JOSAB.445791.

ZHANG, Z. Microsoft Kinect Sensor and Its Effect. **IEEE MultiMedia**, v. 19, n. 2, pp. 4-10, 2012. DOI: https://doi.org/10.1109/MMUL.2012.24.