UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Felipe Zago Canal

# RECONHECIMENTO DE EXPRESSÕES FACIAIS BASEADO EM REDES NEURAIS CONVOLUCIONAIS PARA APLICAÇÃO NO SISTEMA TUTOR INTELIGENTE MAZK

Araranguá,
2021

Felipe Zago Canal

# RECONHECIMENTO DE EXPRESSÕES FACIAIS BASEADO EM REDES NEURAIS CONVOLUCIONAIS PARA APLICAÇÃO NO SISTEMA TUTOR INTELIGENTE MAZK

Trabalho de Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.
**Orientador:** Prof. Antônio Carlos Sobieranski, Dr.
**Coorientadora:** Profa. Eliane Pozzebon, Dra.

Araranguá,
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Canal, Felipe Zago
Reconhecimento de Expressão Faciais Baseado em RedesNeurais Convolucionais para Aplicação no Sistema TutorInteligente MAZK / Felipe Zago Canal ; orientador, Antônio Carlos Sobieranski, coorientadora, Eliane Pozzebon, 2021.
32 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2021.

Inclui referências.

1. Engenharia de Computação. 2. Visão Computacional. 3. Sistema Tutor Inteligente. 4. Reconhecimentode Expressão Facial. 5. Redes Neurais Convolucionais. I. Sobieranski, Antônio Carlos. II. Pozzebon, Eliane. III. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. IV. Título.

Felipe Zago Canal

**RECONHECIMENTO DE EXPRESSÕES FACIAIS BASEADO EM REDES NEURAIS CONVOLUCIONAIS PARA APLICAÇÃO NO SISTEMA TUTOR INTELIGENTE MAZK**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação, e foi aprovado em sua forma final pelo Curso de Engenharia de Computação.

Araranguá, 14 de maio de 2021.


Documento assinado digitalmente
Fabricio de Oliveira Ourique
Data: 21/05/2021 09:50:02-0300
CPF: 916.167.860-00
Verifique as assinaturas em https://v.ufsc.br

**Prof. Fabrício de Oliveira Ourique, Dr.**
Coordenador do Curso

**Banca Examinadora:**

Documento assinado digitalmente

Antonio Carlos Sobieranski
Data: 20/05/2021 21:31:10-0300
CPF: 005.305.809-77
Verifique as assinaturas em https://v.ufsc.br

---

**Prof. Antônio Carlos Sobieranski, Dr.**
Orientador
Universidade Federal de Santa Catarina

Documento assinado digitalmente

Eliane Pozzebon
Data: 21/05/2021 07:59:43-0300
CPF: 779.213.139-49
Verifique as assinaturas em https://v.ufsc.br

---

**Profa. Eliane Pozzebon, Dra.**
Coorientadora
Universidade Federal de Santa Catarina

Documento assinado digitalmente

Anderson Luiz Fernandes Perez
Data: 21/05/2021 07:09:59-0300
CPF: 020.803.459-58
Verifique as assinaturas em https://v.ufsc.br

---

**Prof. Anderson Luiz Fernandes Perez, Dr.**
Avaliador
Universidade Federal de Santa Catarina

Documento assinado digitalmente

Alexandre Leopoldo Goncalves
Data: 20/05/2021 22:17:31-0300
CPF: 786.740.349-34
Verifique as assinaturas em https://v.ufsc.br

---

**Prof. Alexandre Leopoldo Gonçalves, Dr.**
Avaliador
Universidade Federal de Santa Catarina

---

**Prof. Fábio Rodrigues De La Rocha, Dr.**
Avaliador Suplente
Universidade Federal de Santa Catarina

# Reconhecimento de Expressões Faciais Baseado em Redes Neurais Convolucionais para Aplicação no Sistema Tutor Inteligente MAZK

## *Facial Expression Recognition Based on Convolutional Neural Networks for Application in the Intelligent Tutoring System MAZK*

Felipe Zago Canal * Antônio Carlos Sobieranski † Eliane Pozzebon ‡

2021, Maio

## Resumo

A utilização de Sistema Tutores Inteligentes (STIs) não é novidade e vem crescendo a cada dia, principalmente em um contexto de pandemia, onde o ensino remoto assume o protagonismo da educação. Os avanços tecnológicos, principalmente no âmbito das Redes Neurais Convolucionais (CNNs), permitem a criação de ferramentas para esses sistemas, com o intuito de obter melhores resultados de aprendizagem e, ao mesmo tempo, aperfeiçoar sua interação com os estudantes. A partir disso, o presente estudo propõe o desenvolvimento de um modelo de classificação de expressões com base em imagens da face. Esse recurso é, posteriormente, aplicado no STI MAZK, como meio de identificação afetiva dos estudantes durante o processo de ensino-aprendizagem. O modelo proposto foi projetado na estrutura de uma CNN, treinado com mais de 34 mil imagens de dois *datasets*, avaliado e aplicado no tutor. Além disso, foi executada uma análise de utilização de recursos por parte do modelo, onde foi comprovada a viabilidade da sua aplicação junto ao STI. O modelo foi capaz de alcançar uma precisão de média de 98% na classificação de sete expressões distintas, demonstrando-se superior aos métodos aplicados ao mesmo cenário na literatura.

**Palavras-chaves**: Visão Computacional. Sistema Tutor Inteligente. Reconhecimento de Expressão Facial. Redes Neurais Convolucionais.

*felipe.canal@grad.ufsc.br
†a.sobieranski@ufsc.br
‡eliane.pozzebon@ufsc.br

# Reconhecimento de Expressões Faciais Baseado em Redes Neurais Convolucionais para Aplicação no Sistema Tutor Inteligente MAZK

# *Facial Expression Recognition Based on Convolutional Neural Networks for Application in the Intelligent Tutoring System MAZK*

Felipe Zago Canal * Antônio Carlos Sobieranski † Eliane Pozzebon ‡

2021, Maio

## Abstract

Intelligent Tutoring Systems (ITSs) are not a novelty and have been increasing in usage every day, especially in the pandemic context, in which the remote teaching takes the leading role in education. The technological advances, mainly in the scope of Convolutional Neural Networks (CNNs), allow the creation of tools for these systems, aiming better learning results and, at the same time, improving students interactions. Therefore, the present study propose the development of a facial expression classification model. This model is, subsequently, applied to the MAZK ITS, as a tool for students emotion recognition during the learning process. The proposed model was designed as a CNN, trained with more than 34 thousand images from two datasets, evaluated and applied to the tutor. In addition, a resource usage analysis was conducted, proving the viability of its implantation along with the ITS. The model was capable of reaching a precision of 98% in the classification of seven distinct expressions, proving to be superior to the existing methods applied to the same scenario in the literature.

**Key-words**: Computer Vision. Intelligent Tutoring System. Facial Expression Recognition. Convolutional Neural Network.

*felipe.canal@grad.ufsc.br
†a.sobieranski@ufsc.br
‡eliane.pozzebon@ufsc.br

# 1 Introdução

Nos últimos anos, com o evidente avanço e difusão das tecnologias de informação, têm-se cada vez mais espaço para o emprego de ferramentas digitais de auxílio para a educação, como os Sistemas Tutores Inteligentes (STIs)(HWANG, 2003). Esses sistemas caracterizam-se como ambientes virtuais de aprendizagem que incorporam modelos computacionais de áreas como ciências cognitivas, ciências da educação, inteligência artificial, entre outras (GRAESSER; CONLEY; OLNEY, 2012). Desta forma, STIs podem também fazer uso de alguma técnica de Inteligência Artificial (IA) com o objetivo de otimizar o processo de ensino-aprendizagem, e personalizar a interação humano-computador (GALAFASSI *et al.*, 2020).

Para Pozzebon *et al.* (2008), um STI é um sistema para ensino capaz de tomar algum tipo de decisão autônoma de maneira *on-line*, baseado em suas interações com o aluno e, para isso, pode acessar diversos meios de conhecimento para suportar tal decisão. A eficácia desses sistemas é, por vezes, questionada devido à falta de interação afetiva do tutor com o aprendiz, contudo, são capazes de gerar bons resultados de aprendizagem (QI-RONG, 2010). Por mais que os STIs sejam capazes de promover um aprendizado personalizado, com conteúdo otimizado ao perfil individual do estudante, seu impacto pode ser afetado uma vez que o aprendizado humanizado é deixado de lado e a falta de emoções pode causar resultados menos significativos, quando comparados ao ensino tradicional (WU; LIU; WANG, 2008). Portanto, um STI ideal deveria focar não somente no conhecimento, mas também nas emoções do aprendiz (AKPUTU; SENG; LEE, 2013).

O STI MAZK, desenvolvido no Laboratório de Tecnologias Computacionais (LabTeC) da Universidade Federal de Santa Catarina (UFSC), campus Araranguá, é um tutor inteligente para ensino e aprendizagem de diversos temas. No MAZK é identificado e armazenado o nível de conhecimento do usuário, assim como as dificuldades dos exercícios que são ajustados conforme a interação do aluno com o tutor. O software promove aprendizagem de forma adaptativa e colaborativa e pode ser acessado em computadores ou a partir de outros dispositivos, como celulares ou *tablets* (BITTENCOURT *et al.*, 2018).

No MAZK, o professor é capaz de acompanhar alunos durante o processo de aprendizagem de forma síncrona através do recurso de sala virtual. Nesse ambiente, o professor pode interagir com os estudantes por meio de *chat* de texto ou até mesmo criando uma videoconferência. Todas as informações de interações dentro do MAZK são armazenadas nos modelos do aluno, pedagógico e de domínio (SILVA *et al.*, 2019; MORO *et al.*, 2019). Contudo, fora do ambiente de video conferência, as emoções que os estudantes de fato expressam não estão ao alcance do professor, tampouco do tutor. Uma forma de preencher essa lacuna é pela extração da emoção da face do aprendiz a partir de uma imagem adquirida com câmera do dispositivo no qual está sendo realizada a interação.

O reconhecimento de expressões pela face é uma tarefa relativamente simples e realizada e maneira natural por humanos, contudo ainda é uma área em aberto no meio computacional (BISWAS; SIL, 2015). Apesar de não se ter acesso a um método com efetividade comparada à humana para realização de tal tarefa, são vários os algoritmos e ferramentas aptos a categorizar emoções a partir de imagens faciais e obter alto índice de corretude. Alguns métodos de reconhecimento de expressão pela face são aplicados a STIs a certo tempo, todavia, devido ao seu custo computacional, são usualmente bem simplificados e superficiais, baseando-se em poucas métricas para realização das classificações (AMMAR *et al.*, 2010; LIN *et al.*, 2012; BALDASSARRI *et al.*, 2015; WU; LIU; WANG, 2008).

Por outro lado, com o progresso do poder computacional, principalmente das unidades de processamento gráfico (GPUs), problemas que envolvem o processamento de imagens, antes limitados por seu alto custo computacional, têm sua solução alcançada (HAENSCH; GOKMEN; PURI, 2018), o que abre uma nova janela de opções para desenvolvimento e aplicação de sistemas mais complexos, porém precisos em diversas áreas, dentre elas, os STIs. Na literatura, a obtenção da expressão do estudante é um fator positivo para um STI e pode agregar valor ao processo de ensino-aprendizagem, no entanto, realizar a extração dessa informação em um sistema tutor, não é uma função fácil, principalmente visto que isso precisa ocorrer em um curto período de tempo e sem afetar o desempenho do sistema. Recentemente, foram verificados trabalhos apresentando diversos algoritmos com eficácia comprovada para extração de expressões de imagens da face (ALI *et al.*, 2015; SLIMANI *et al.*, 2018; PUTHANIDAM; MOH, 2018; HAPPY; ROUTRAY, 2014; LOPES *et al.*, 2017; GAN, 2018; MOHSENI; ZAREI; RAMAZANI, 2014; HU *et al.*, 2019), alguns deles podendo atingir 100% de acurácia quando avaliados sobre determinado conjunto de imagens, de domínio bastante controlado.

Neste trabalho é proposta uma solução computacional para uso de reconhecimento de emoções em expressões faciais, com enfoque em um sistema tutor inteligente. Tendo em vista as necessidades do sistema, bem como as limitações do ambiente de execução e os métodos disponíveis para extração das emoções, um modelo de Rede Neural Convolucional (CNN) é proposto e aplicado para o reconhecimento de emoções a partir de imagens no sistema tutor inteligente MAZK. O treinamento do modelo em questão é realizado de maneira supervisionada e com a inserção de dois *datasets* na rede: CK+ e MUG. As métricas de validação do modelo apontam uma precisão média quantitativa de 98% para a detecção das seguintes emoções: neutro, raiva, felicidade, nojo, medo, tristeza e surpresa. Além disso, uma aplicação experimental preliminar do modelo obtido é feita no MAZK com o objetivo de validar sua aplicabilidade, sem comprometer o funcionamento do STI.

O restante deste trabalho é organizado da seguinte forma: Alguns termos e conceitos essenciais para o trabalho são apresentados na Seção 2. Uma breve apresentação do Sistema Tutor Inteligente MAZK é conduzida na Seção 3. Logo após, na Seção 4, são apresentados os trabalhos correlatos pertinentes a pesquisa realizada na literatura sobre sistemas de reconhecimento de expressão pela face aplicados em STIs. Na sequência, são apresentados a metodologia de desenvolvimento do modelo proposto e sua aplicação na ferramenta (Seção 5). Após isso, são apresentados os resultados obtidos, tanto na avaliação da rede, quanto na sua aplicação no MAZK (Seção 6) e, por fim, são discutidas propostas de aplicações futuras e o trabalho é concluído na Seção 7.

# 2 Fundamentação Teórica

## 2.1 Aprendizado de Máquina

O aprendizado de máquina, que diz respeito à detecção de qualquer tipo de padrão significativo em um conjunto de dados, se tornou uma ferramenta muito utilizada por diversas áreas nas últimas décadas, desde a bioinformática até a astronomia (SHALEV-SHWARTZ; BEN-DAVID, 2014). Devido ao grande aumento na quantidade de informação armazenada digitalmente, o aprendizado de máquina se faz necessário para que se possa desenvolver métodos de detecção automática de padrões nessas informações e, com isso, ressaltar aquilo que for de maior interesse para determinada aplicação (MURPHY, 2012).

Os dois principais tipos de aprendizado são conhecidos por aprendizado supervisio-

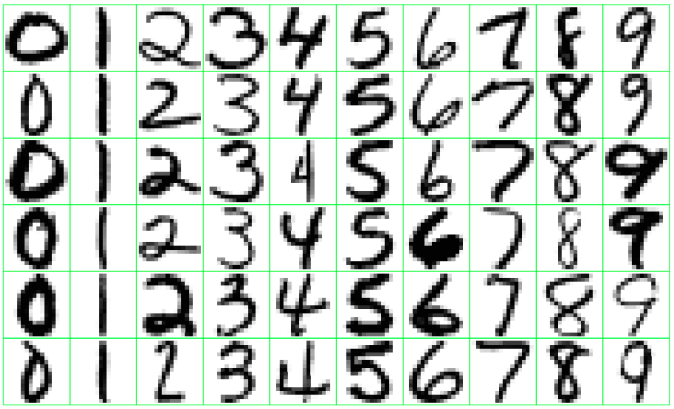
nado e não supervisionado. O aprendizado supervisionado recebe tal nome por conta da presença de uma variável de saída que guia o processo de aprendizado (HASTIE; TIBSHIRANI; FRIEDMAN, 2009), ou seja, durante a etapa de treinamento, o classificador recebe junto com a entrada, um valor esperado de saída, também conhecido como rótulo. Por outro lado, o aprendizado não supervisionado não recebe rótulo no processo de treinamento e, portanto, tem o objetivo de descrever as associações e padrões presentes no conjunto de dados.

O tipo de aprendizado aplicado nesse trabalho é o supervisionado, tendo em vista o *dataset* a ser utilizado, o qual é composto por pares de imagem e rótulo. Por via de regra, o algoritmo de aprendizado supervisionado tem como verdadeiro e correto todo o rótulo que acompanha um dado de entrada e, por consequência, a qualidade e tamanho do conjunto de dados de treinamento é crucial para o bom funcionamento do método no momento de predição (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018).

O método de classificação pode ser entendido como uma função que, ao receber uma determinada entrada, produz uma saída específica. Com o processo de aprendizado supervisionado, o classificador, ao receber um conjunto de dados juntamente com a saída esperada para cada uma das amostras, é capaz de alterar sua representação dessa função, podendo então, produzir a saída esperada para todos ou a maioria dos valores de entrada.

Um exemplo de aplicação de aprendizado de máquina, mais especificamente, supervisionado é o reconhecimento de dígitos escritos a mão, conforme apresentado por Hastie, Tibshirani e Friedman (2009). O *dataset* desse problema é composto por códigos de CEP da caixa postal dos Estados Unidos escritos à mão. Cada imagem contém um número pertencente ao código de cinco caracteres e cada amostra é dimensionada em 16x16 *pixels* com valores de preto e branco que variam de 0 a 255. Algumas amostras podem ser visualizadas na Figura 2.1.

Figura 1 – Amostras de dígitos escritos à mão para o problema de reconhecimento de CEP da caixa postal dos Estados Unidos.



Fonte: (HASTIE; TIBSHIRANI; FRIEDMAN, 2009)

Com a obtenção de bons resultados na classificação dessas imagens, o algoritmo poderia ser integrado a um processo automatizado de ordenação dos envelopes.

## 2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNNs) têm obtido resultados revolucionários em diversas áreas relacionadas ao reconhecimento de padrões na última década e, para tal, dois aspectos importantes desse tipo de rede são: a redução do número de parâmetros de uma Rede Neural Artificial (RNA), e a capacidade de extrair características abstratas à medida que a entrada se propaga pelas camadas da rede (ALBAWI; MOHAMMED; AL-ZAWI, 2017). CNNs são uma configuração de RNA usada para o processamento de dados que possuem um padrão matricial (duas dimensões) e que se baseiam, tipicamente, em três camadas: convolução, *pooling* e densa (YAMASHITA *et al.*, 2018).

### 2.2.1 Convolução

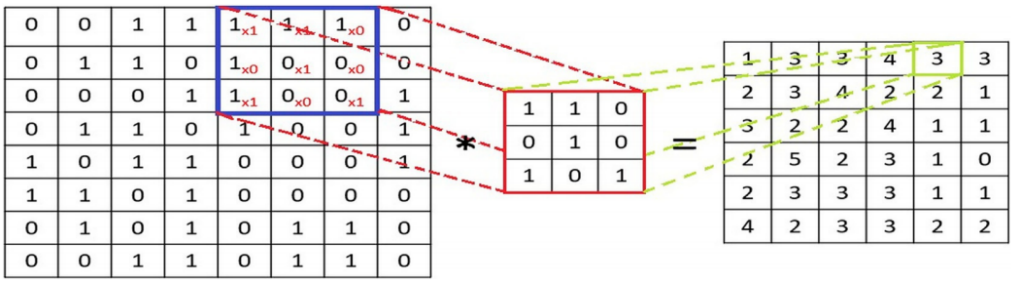
Convolução é um tipo específico de operação linear usado para extração de características, onde uma pequena matriz, também conhecida por *kernel* (k) é aplicada ao longo da entrada (YAMASHITA *et al.*, 2018). A representação da convolução entre a imagem de entrada (I) e o kernel (K) ocorre conforme mostrado na Equação 1.

$$(K * I)(x, y) = \sum_{l=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} K(l, j) I(x + l, y + j) \tag{1}$$

onde $m$ e $n$ são o número de linhas e colunas de $K$, respectivamente e $x$ e $y$ as coordenadas do ponto analisado de I.

O kernel que, costumeiramente para problemas de reconhecimento de padrões em imagens, assume dimensão 3x3 ou 5x5 (WANG; LIN; WANG, 2016), é aplicado ao longo de toda a extensão da imagem, deslizando sobre os *pixels* e produzindo o mapa de ativação (SINGH; MEITEI; MAJUMDER, 2020), conforme demonstrado na Figura 2.

Figura 2 – Demostração da convolução entre o *kernel* e uma porção da imagem, pela aplicação da Equação 1.



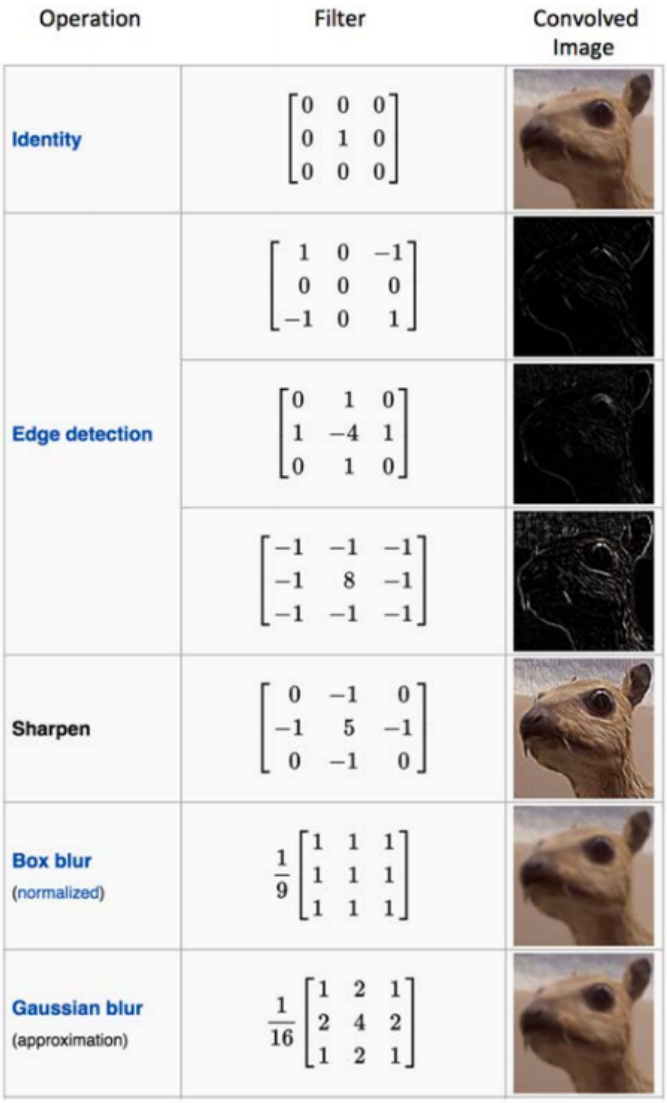

Fonte: (SINGH; MEITEI; MAJUMDER, 2020)

Uma das caraterísticas importantes do processo de convolução em CNNs é a redução na quantidade de dados a serem propagados pela rede. Como exemplificado na Figura 2, tomando uma imagem de 8x8 como entrada e usando um *kernel* de 3x3 *pixels*, ao deslizar-se $k$ por toda a matriz de entrada não é possível atingir os *pixels* mais extremos da imagem e, portanto, obtém-se uma saída de 6x6 *pixels*.

O processo de convolução pode ser também interpretado como a aplicação de um filtro de extração de características da imagem para obter informações como bordas, por

exemplo. Sendo o filtro o próprio *kernel*, este pode assumir diversos valores e, dessa forma, produzir uma saída específica (Figura 3).

Figura 3 – Resultado da aplicação de distintas configurações de *kernel* sobre uma imagem.




Fonte: (ALBAWI; MOHAMMED; AL-ZAWI, 2017)

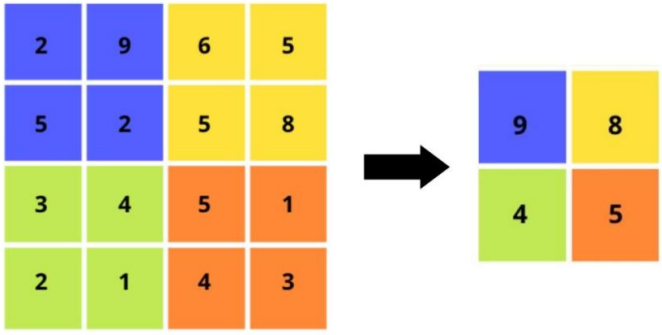
Para as redes neurais convolucionais, a aplicação de uma função de ativação ao final de cada camada convolutiva se faz necessária. Essa função tem papel muito importante no processo, sendo responsável por definir a ativação ou não de cada neurônio presente na camada e, sendo assim, ditando quais características são levadas adiante na rede. Funções como sigmoide e tangente hiperbólica são usualmente empregadas para essa tarefa (LI *et al.*, 2014), porém funções não lineares, como a *rectified linear function* (ReLU) tem se tornado muito populares (RAWAT; WANG, 2017) por conta de seu desempenho.

### 2.2.2 *Pooling*

Diferentemente da camada convolutiva, a operação de *pooling* tem como único e específico objetivo, a redução de dimensionalidade das características extraídas da camada anterior. No caso das CNNs, a camada de *pooling* geralmente segue a camada de convolução para condensar o mapa de características produzido (GUO *et al.*, 2016). Os algoritmos de *pooling* mais utilizados são *max-pooling* e *average-pooling* onde são escolhidos o *pixel* com maior valor dentre a seção anualizada e a média da seção, respectivamente (LI *et al.*, 2014).

Na Figura 4, pode-se observar que ao usar uma matriz de *pooling* de dimensão 2x2, reduz-se a quantidade de informação que é propagada pela rede em 75%, uma vez que se escolhe apenas um *pixel* a cada 4, reduzindo o custo computacional e prevenindo *overfitting* (SINGH; MEITEI; MAJUMDER, 2020) [1].

Figura 4 – Exemplo de aplicação do processo de *pooling* com uma matriz 2x2 e o resultado obtido da operação.




Fonte: Elaborado pelo Autor

### 2.2.3 Camada Densa

A camada densa, também conhecida como completamente conectada, recebe esse nome justamente por conta de sua estrutura onde cada neurônio dessa camada, se conecta com todos os neurônios das camadas prévia e subsequente por meio de um peso sináptico ajustável (YAMASHITA *et al.*, 2018). Essa camada é responsável por operações lógicas de alto nível, coletando as características extraídas da camada prévia e provendo a decisão final da classificação (SINGH; MEITEI; MAJUMDER, 2020).

O principal ponto negativo da camada densa é seu custo computacional elevado, pois envolve muitos parâmetros que inserem complexidade ao treinamento do modelo. Para evitar o uso demasiado de processamento, usualmente insere-se uma camada de *dropout* antes camada densa. Essa camada inativa alguns neurônios aleatórios da rede, evitando o sobre-ajuste, além de reduzir significativamente o custo computacional da rede na etapa de treinamento (ALBAWI; MOHAMMED; AL-ZAWI, 2017; SINGH; MEITEI; MAJUMDER, 2020).

[1] Quando um modelo se ajusta muito bem ao conjunto de dados de treinamento, mas falha muito na previsão de novos dados.

## 2.3 *Tensorflow*

O sucesso no crescimento e difusão do aprendizado de máquina é dado, segundo Abadi *et al.* (2016), pela invenção de modelos mais sofisticados, pela disponibilidade de grandes conjuntos de dados de diferentes áreas e pelo desenvolvimento de plataformas de software que possibilitam a simples arquitetura e treinamento desses novos modelos. Dentre essas plataformas, está o *Tensorflow*, lançado pelo Google em novembro de 2015 (GOLDSBOROUGH, 2016).

*Tensorflow* é um *framework*[2] de código aberto para desenvolvimento de modelos de aprendizado de máquina que oferece uma alta abstração no desenvolvimento de algoritmos, desde o nível iniciante, até estruturas mais complexas, aplicadas por grandes empresas. Um modelo desenvolvido com o auxílio do *tensorflow* pode ser executado em uma alta gama de sistemas, variando de simples dispositivos móveis até grandes sistemas distribuídos com o uso de unidades de processamento gráfico (GPUs), sem necessidade de adaptação.

Além de facilitar o processo de construção e validação de modelos, o *framework* possui uma versão otimizada para processamento em unidades gráficas (GPUs) que pode acelerar muito o processo de treinamento de redes neurais (cerca de 3 vezes mais rápido em relação ao processamento tradicional em CPU (ALZANTOT *et al.*, 2017)).

Adicionalmente ao *tensorflow*, para a execução desse projeto foi utilizada a ferramenta *Keras*, que trata-se de uma biblioteca de código aberto que funciona como uma interface de abstração dos recursos do *framework*. O *Keras* é capaz de facilitar o trabalho do desenvolvedor além de promover uma estrutura simplificada para construção de modelos de rede neural artificial e possuir integração com *softwares* para visualização do processo de aprendizagem do modelo.

Na Figura 5 pode-se visualizar a simplicidade na criação de um modelo de rede neural e a adição de duas camadas densas a ele. Graças ao *tensoflow*, juntamente com a biblioteca *keras*, é possível a criação e validação de modelos de inteligência artificial de maneira fácil e rápida, resolvendo problemas de aprendizado de máquina com alta abstração.

Figura 5 – Construção de um modelo sequencial de rede neural artificial onde são adicionados, com a utilização de *keras* e *tensorflow*, duas camadas densas de neurônios com 64 e 10 elementos, respectivamente. Além disso, é possível ainda especificar o algoritmo de ativação a ser usado em cada uma das camadas independentemente.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

modelo = Sequential()
modelo.add(Dense(units=64, activation='relu'))
modelo.add(Dense(units=10, activation='softmax'))
```

Fonte: Elaborado pelo Autor

[2] Pacote de códigos prontos que podem compreender conjuntos de funções e que pode ser reutilizado para acelerar o desenvolvimento de aplicações.

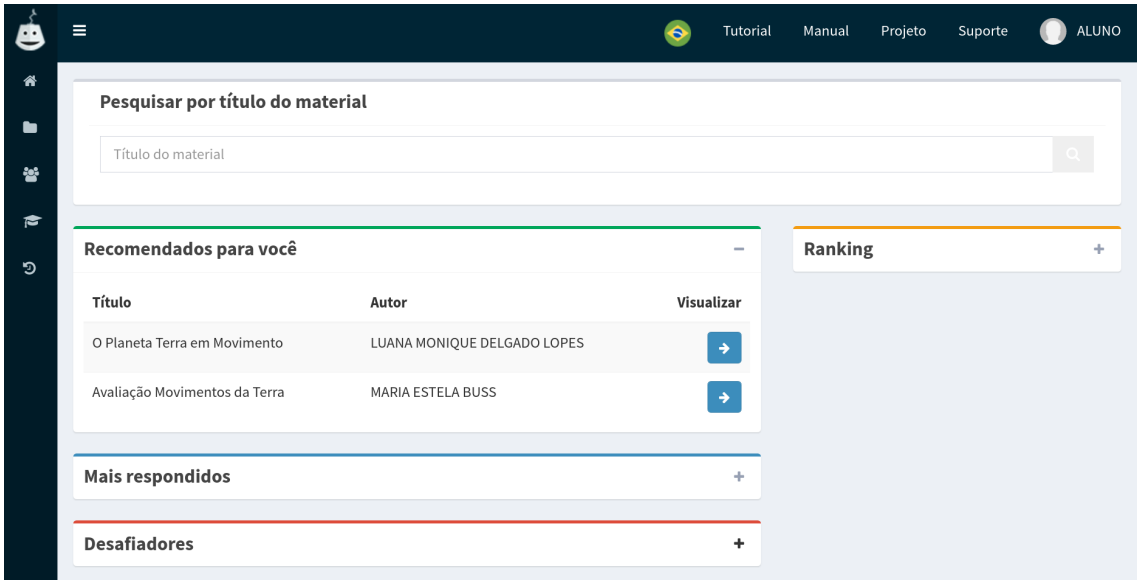
# 3 Sistema Tutor Inteligente MAZK

O MAZK é um sistema tutor inteligente desenvolvido pela equipe do Laboratório de Tecnologias Computacionais (LabTeC) da Universidade Federal de Santa Catarina - Campus Araranguá. Além de se apropriar de conceitos de Inteligência Artificial para se tornar um facilitador do aprendizado, o MAZK se posiciona como um instrumento de apoio pedagógico para as estratégias de ensino do professor (BITTENCOURT *et al.*, 2018).

A arquitetura multiagentes adotada pelo sistema visa a integração de seres humanos com entidades artificiais para proporcionar colaboratividade e dar suporte à aprendizagem (VIDOTTO *et al.*, 2017). Os agentes empregados nesse processo são capazes de identificar os níveis de saber do utilizador, bem como ajustá-los de acordo com as interações entre o aluno e o ambiente, levando em conta a dificuldade dos conteúdos (JOSUÉ *et al.*, 2018).

O acesso ao MAZK pode ser feito por três tipos de usuários: aluno, professor e coordenador. O aluno, ao cadastrar-se e iniciar as interações com o sistema, é apresentado a diversos conteúdos que, ao longo do tempo são adaptados pelo tutor ao seu perfil (Figura 6).

Figura 6 – Demonstração das recomendações de materiais feitas pelo tutor ao aluno, onde pode-se perceber também a apresentação dos materiais mais estudados e de desafios propostos ao estudante.




Fonte: (MAZK, 2021)

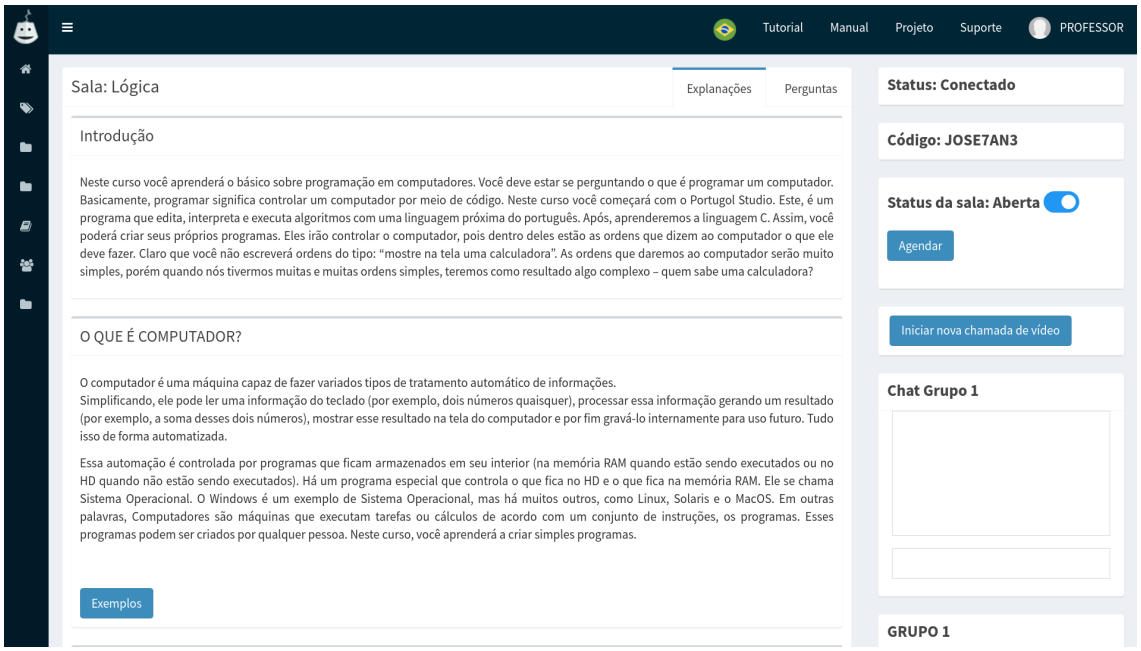
Os professores que se cadastram no MAZK, após comprovação de docência, são autorizados a inserir novos materiais na ferramenta, disponibilizando-os publicamente para que outros professores possam utilizá-los em suas salas e ampliando o acervo de conteúdos abrangidos pelo tutor. Além disso, os usuários coordenadores têm a possibilidade de criar cursos em formatos de módulos, onde podem ser abordados diversos temas, disponibilizando os conteúdos dos materiais próprios ou públicos do sistema.

Os materiais criados pelos professores podem ser compostos por explanações, exemplos e perguntas (objetivas ou descritivas), sejam elas de autoria própria ou provindas do acervo do STI. Além disso, o sistema possibilita o ensino e aprendizagem de diversos

temas e conteúdos de forma adaptativa e colaborativa, tanto por parte do professor quanto dos estudantes. A colaboratividade do sistema é ressaltada na utilização do recurso de salas. Esses ambientes podem ser criados por tutores humanos, sendo que, no ato da sua geração, um código é informado e este pode ser compartilhado entre os alunos para que obtenham acesso ao ambiente.

Dentro das salas, os alunos podem ser divididos em grupos e apresentados a um material para que, de maneira coletiva, realizem o estudo do conteúdo e a resolução dos problemas propostos. Para a facilitação da interação entre seus membros, cada grupo tem a sua disponibilidade um *chat* onde ideias de resolução dos problemas podem ser compartilhadas. O professor, caso presente na sala virtual durante a atividade, pode se comunicar separadamente com os *chats* de cada grupo de forma textual ou criar uma video conferência com todos os membros da sala (Figura 7).

Figura 7 – Visão do professor do ambiente de sala dentro do MAZK. Pode-se observar a seção de *chat* e opção de abertura de uma video conferência, apesar de ninguém estar presente no momento da captura da imagem.




Fonte: (MAZK, 2021)

Além do recurso de salas, todo e qualquer aluno que se cadastre no sistema, pode usufruir dos materiais públicos disponibilizados pelo tutor, bem como dos cursos oferecidos pelos coordenadores. Ao longo de suas interações com o sistema, o tutor atribui e atualiza o nível de experiência de cada estudante e gera um *ranking* o que, segundo Bittencourt *et al.* (2018), por ser uma estratégia de gamificação, visa promover a interação social entre os estudantes e é capaz de imprimir maior motivação ao aprendizado.

Com a captura de imagens da face dos estudantes durante sua interação com o STI, seja ela na resolução de problemas ou estudo de conteúdos, é possível a detecção de sua emoção e, a partir disso, a avaliação da necessidade de intervenção do docente (MALDANER *et al.*, 2019). No MAZK, essa decisão pode ser designada ao tutor inteligente

e aplicada por um método de *loop interno*[3], onde o tutor interage com o aluno a partir da detecção de determinada emoção, com o intuito de auxiliar o estudante a superar uma dificuldade, por exemplo.

# 4 Trabalhos Correlatos

Em busca na literatura, pode-se encontrar algumas abordagens de sistemas de reconhecimento de emoções por imagens da face aplicados a STIs. Dentre esses trabalhos, destacam-se a aplicação de métodos baseados em distâncias entre pontos estratégicos da face, e os trabalhos baseados em Redes Neurais Artificiais (RNAs) com treinamento utilizando grandes datasets e auxílio de bibliotecas bem conhecidas, como OpenCV, por exemplo. Contudo, a grande maioria dos trabalhos carece de detalhes sobre as implementações propostas, impossibilitando a replicação das técnicas e validação dos resultados.

- Ammar *et al.* (2010) propõem a introdução do conceito de computação afetiva dentro dos STIs. O objetivo principal dos autores é analisar as expressões faciais do estudante e aprimorar a interação entre o mesmo com o sistema, agregando informação ao modelo do aluno e possibilitando um monitoramento mais completo do processo de aprendizagem. A solução apresentada neste trabalho é composta por duas etapas: extração de características da expressão facial e classificação da emoção. Na primeira etapa do processo, a região da face é determinada, bem como a localização do contorno dos olhos, sobrancelhas e boca. Os algoritmos usados foram baseados na forma geométrica do rosto, porém, não são detalhados pelos autores. Por outro lado, a classificação das expressões é realizada a partir da análise de seis distâncias entre pontos estratégicos da face (olhos, boca e sobrancelhas).

  Esse algoritmo pode ser executado de maneira muito simples e rápida, contudo requer a disponibilidade de uma imagem base, referente à expressão neutra, para efetuar a classificação. Segundo os autores, o algoritmo alcançou uma acurácia de 80% na classificação das expressões de raiva, nojo, medo, felicidade, tristeza e surpresa. As predições do algoritmo são salvas no modelo do aluno, possibilitando que os agentes do sistema façam uso dessa informação para adotar uma estratégia de ensino específica.

- Lin *et al.* (2012) trazem a computação afetiva para os STIs, assim como Ammar *et al.* (2010), porém, além da implantação de reconhecimento de emoções a partir da face, os autores visam extrair esse mesmo tipo de informação dos textos escritos pelos aprendizes. Para a análise das imagens, essa aplicação realiza a extração de informações pelo uso de um algoritmo próprio, baseado em *Active Shape Model* (ASM), o qual é aplicado diretamente sobre imagens da *webcam* dos estudantes. A partir disso, pontos específicos são extraídos dessas imagens e submetidos ao mesmo método de classificação proposto por Ammar *et al.* (2010).

  Os dados de emoção, extraídos das imagens dos usuários do sistema, são utilizados pelo STI para prover dois tipos de *feedback*: auditivo e de animação, por exemplo, se a imagem do aluno for classificada como "triste", o tutor deve confortá-lo mostrando uma feição carinhosa e perguntando se está tudo bem.

[3] *Loop* interno é um dos dois *loop* definidos por Vanlehn (2006) que caracteriza as etapas de interação do aluno com o tutor, podendo prover *feedback* ou dicas ao estudante.

- Zatarain-Cabada *et al.* (2015) apresenta a implementação de um STI para o ensino da linguagem de programação JAVA onde são empregados dois algoritmos de computação afetiva, um para reconhecimento de emoções em texto e outro em imagens. A implementação proposta pelos autores para a interpretação das imagens é baseada em uma RNA. Inicialmente, um extrator de características (não especificado) é implementado pela biblioteca OpenCV e, então, essas características são submetidas à RNA (implementada no software WEKA®) para o reconhecimento das seguintes emoções: alegria, surpresa, tristeza, raiva e neutra.

  O sistema faz uso de um conjunto de regras *fuzzy* para definir o nível de complexidade do problema a ser resolvido pelo estudante na sequência.

- Baldassarri *et al.* (2015) descreve a criação de uma plataforma de tutoria para *Interactive Digital TeleVision* (IDTV). A característica principal deste ambiente é que o aluno realiza interações com o sistema a partir de uma televisão, e não de um computador. Contudo, a abordagem para reconhecimento de emoções faciais não se altera, uma vez que, do ponto de vista do algoritmo, o que importa são as imagens apresentadas como entrada. O sistema parte de uma análise de 20 pontos específicos do rosto (sobrancelhas, olhos e boca) e é capaz de classificar a emoção expressa pelo estudante, tendo como base as distâncias desses pontos em relação aos seus equivalentes na face neutra. Segundo os autores, para as expressões de raiva, nojo, medo, felicidade, tristeza e surpresa, o algoritmo alcança uma acurácia de 84,92%.

- Em uma abordagem de desenvolvimento de um STI para ensino de matemática ao terceiro ano, Barrón Estrada, Zatarain Cabada e Hernández Pérez (2014) propuseram uma ferramenta com reconhecimento de expressões que retorna um *feedback* ao aluno enquanto ele desenvolve atividades no tutor. O reconhecimento das emoções é feito por uma RNA que, por sua vez, é treinada com o *dataset* RaFD (LANGNER *et al.*, 2010). Apesar do *dataset* utilizado conter imagens de 8 emoções, o tutor classifica as 6 emoções básicas (raiva, nojo, medo, felicidade, tristeza e surpresa) além da expressão neutra. A rede é desenvolvida em Java e C++ com as bibliotecas JavaCV e OpenCV, respectivamente, e os resultados são considerados muito satisfatórios, pelos próprios autores, contudo, nenhum dado numérico de precisão é apresentado.

- Com o intuito de agregar o componente emocional em STIs tradicionais, os autores Wu, Liu e Wang (2008) propõem a criação de um modelo baseado no reconhecimento de emoções, tanto pela imagem da face quando por identificações textuais. O método de classificação das emoções pela imagem da face é baseado na localização de três pontos faciais, relativos às sobrancelhas e boca. Um ângulo entre os três pontos é calculado e, a partir disso, os autores classificam a expressão do indivíduo como calmo, feliz ou triste.

A partir da análise dos métodos presentes na literatura atualmente, pode-se perceber uma diversidade nas emoções consideradas por cada trabalho. Contudo, a maioria dos sistemas de reconhecimento de expressões é baseada nas seis expressões básicas de Ekman e Friesen (1971), são elas: raiva, nojo, medo, felicidade, tristeza e surpresa. Tais expressões, juntamente com a neutra, foram consideradas também neste trabalho.

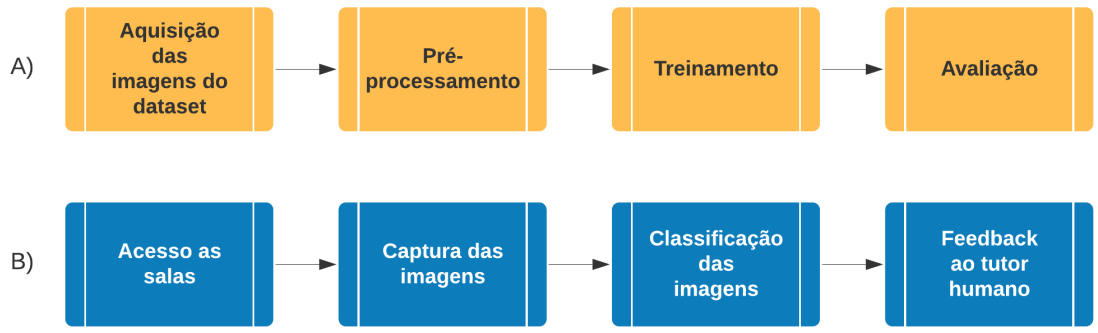
Além disso, percebe-se uma clara tendência de uso de algoritmos muito simples, muitas vezes escolhidos por conta do custo computacional requerido para sua execução e tendo em vista que, no cenário de STIs, o recurso é limitado e precisa ser compartilhado

com outros módulos dos sistemas. Contudo, com o avanço da tecnologia, muitos algoritmos já foram otimizados e adaptados para suas versões com execução paralela alcançando performance próximas a tempo real (TURABZADEH *et al.*, 2017). Esses avanços possibilitam arquitetar uma solução mais completa e baseada em métricas de eficácia comprovada para o reconhecimento de emoções, sem prejudicar o fluxo do aprendizado em STIs modernos.

# 5 Método Proposto

Os processos propostos neste trabalho são ilustrados na Figura 8, onde 2 entidades principais podem ser visualizadas: (A) construção de modelo de CNN para o reconhecimento de emoções/expressões faciais e (B) utilização do modelo no STI MAZK. Na Figura 8-A, pode-se observar a obtenção das imagens seguida da etapa de pré-processamento para somente então partir para o treinamento propriamente dito e, por fim, para a validação do modelo. Para a utilização do modelo por parte do MAZK, na Figura 8-B pode-se constatar que, por meio do acesso as salas do tutor, parte-se para a captura das imagens dos estudantes para posterior classificação e *feedback* da expressão ao tutor humano.

Figura 8 – Demonstração dos fluxos das duas principais entidades metodológicas do trabalho.




Fonte: Elaborado pelo autor

## 5.1 Ambientes

Para o desenvolvimento e aplicação desse projeto foram utilizados dois ambientes: (i) o ambiente de desenvolvimento, onde foram executadas todas as etapas de preparação do *dataset*, construção do modelo e treinamento do mesmo, e (ii) o ambiente de produção, onde o reconhecimento de expressões é aplicado de fato.

O ambiente de desenvolvimento, onde foram executadas todas as etapas, desde aquisição do *dataset*, até o treinamento da CNN, foi composto por um processador Intel® Core™ i5-8600K com frequência base de 3.6 GHz e 6 núcleos de processamento, acompanhado de 16 GB de memória RAM de 2666 Hz. Além disso, visto que o processo de treinamento do modelo proposto pode ser muito custoso e tendo em vista a utilização da biblioteca *Tensorflow*, a qual possui versão otimizada para execução paralela em GPUs, utilizou-se uma NVIDIA GeForce GTX 1060 com 6 GB de memória GDDR5 de 8008 MHz e 1280 núcleos CUDA.

O ambiente de produção, onde a CNN é aplicada no STI é o próprio servidor da aplicação, onde foi criado um micro serviço responsável pelo preprocessamento das imagens e posterior classificação através do modelo. Essa máquina é composta por quatro núcleos do processador Intel(R) Xeon(R) CPU E7-4830 v2 @ 2.20 GHz e 6 GB de memória RAM.

## 5.2 *Dataset*

Visto que o desempenho da rede é altamente dependente do conjunto de dados a ela apresentados na etapa de treinamento, uma boa escolha do *dataset* é muito importante. Para o desenvolvimento deste trabalho, foram utilizados dois *datasets*: *Extended Cohn-Kanade Dataset* (CK+) e *MUG Facial Expression Database* (MUG).

O *dataset* CK+ (LUCEY *et al.*, 2010), é uma base bem estabelecida na literatura e amplamente usada desde sua primeira versão Kanade, Cohn e Tian (2000). O conjunto de dados é composto por 593 sequências de imagens de 123 diferentes indivíduos, dos quais 69% são mulheres, 31% homens, 81% Euro-Americanos, 13% Afro-Americanos e 6% de outros grupos. As sequências mencionadas são grupos de imagens que partem da expressão neutra e se alteram gradualmente até um pico de uma determinada emoção, a qual é categorizada em relação ao *Facial Action Coding System* (FACS) [4], e recebem um rótulo referente a uma das seguintes expressões: neutra, alegria, surpresa, tristeza, nojo, raiva ou medo. As amostras são disponibilizadas em resolução 640x470 ou 640x480 *pixels*.

O *dataset* MUG (AIFANTI; PAPACHRISTOU; DELOPOULOS, 2010), por outro lado, consiste em imagens de 86 indivíduos brancos, sendo 35 mulheres e 51 homens com idades de 20 a 35 anos. Contudo, as imagens de apenas 52 participantes são disponibilizadas pela internet, perante solicitação. Esse conjunto é formado por 1462 sequências de imagens e as expressões compreendidas são: neutra, raiva, nojo, medo, felicidade, tristeza e surpresa. Diferentemente do *dataset* CK+, as sequências presentes nessa base partem de uma expressão neutra, chegam a um pico da emoção determinada e voltam ao estado neutro. As amostras possuem resolução de 896x896 *pixels* e, no todo, chegam a atingir 38 GB de dados.

Neste trabalho foram consideradas as expressões convergentes entre os dois *datasets* utilizados, ou seja, o modelo foi desenvolvido para classificar as seguintes expressões: neutra, raiva, felicidade, nojo, medo, tristeza e surpresa.

## 5.3 Modelo de CNN proposto

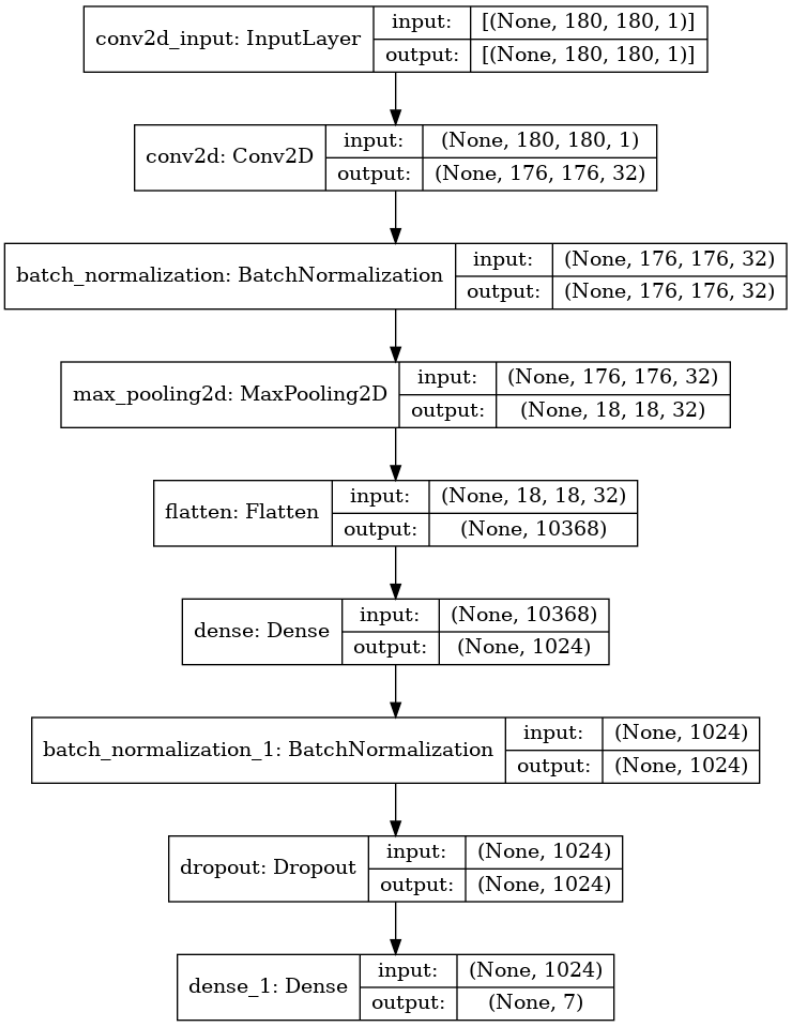
O modelo desenvolvido para classificação de expressões a partir de imagens da face foi estruturado com base nas principais camadas comumente presentes em CNNs, conforme apresentado na subseção 2.2: convolução, *pooling* e camada densa. Todavia, algumas camadas adicionais foram integradas ao modelo para aprimorar a capacidade de classificação, como normalização, linearização e *dropout*, por exemplo.

A camada de normalização (*Batch Normalization*) aplica uma transformação nos dados para manter a média da saída próxima de zero e o desvio padrão de saída, próximo de 1. A camada de linearização simplesmente concatena as colunas de *pixels* uma após a outra, transformando a entrada de duas dimensões que representa os *pixels* da imagem em uma lista, facilitando a propagação da informação ao longo da rede e possibilitando a

[4] Sistema de codificação do movimento da musculatura facial em *Action Units*, *Action Descriptors* e *Movements*. Proposto por (FRIESEN; EKMAN, 1978), o sistema ainda pode ser usado com combinações dos códigos e atribui graus de intensidade para cada um deles.

aplicação de outras camadas no modelo. Por fim, a camada de *dropout* é aplicada para realizar a eliminação aleatória de um subgrupo da informação de entrada, evitando que a rede atinja um estado de sobre-ajuste. O modelo completo é apresentado na Figura 9.

Figura 9 – Estrutura da rede neural convolucional proposta onde cada caixa representa uma camada na rede e é executada de forma sequencial, partindo da entrada de duas dimensões (180x180 *pixels*) até a saída como uma lista de 7 elementos, um para cada expressão considerada.




Fonte: Elaborado pelo autor

### 5.3.1 Pré-processamento

A primeira etapa do processo de desenvolvimento do modelo ocorreu com a preparação dos dados de treinamento da rede neural. Os dois *datasets* selecionados para essa tarefa são disponibilizados de maneiras distintas, apesar de ambos entregarem sequências de imagens que representam transições entre expressões. Enquanto a base CK+ possui sequências de imagens que partem da expressão neutra e se encerram no pico da determinada expressão, o conjunto de imagens da base MUG possui mais uma transição, retornando a

expressão neutra. Além disso, a base CK+ não possui sequências específicas da expressão neutra, diferentemente do *dataset* MUG.

Tendo em vista tais características de ambos os conjuntos, a seleção das imagens a serem consideradas nesse trabalho ocorreu da seguinte forma: Para o *dataset* CK+, o subgrupo inicial de 20% das imagens de cada sequência foi considerado como expressão neutra e os últimos 60% como a expressão relativa a aquele grupo (feliz, por exemplo). Os 20% centrais da divisão foram descartados, visto que, por serem uma mescla das emoções produzidos pela transição, poderiam atrapalhar a classificação da rede.

Para o *dataset* MUG, foram considerados apenas o subgrupo central de cada sequência, onde foram extraídas 20% das imagens por serem essas as mais representativas de cada emoção. Além disso, não se fez necessário extrair amostras de expressão neutra das sequências específicas, pois este *dataset* possui sequências exclusivas desta expressão.

Ao fim do processo, a quantidade de imagens restantes que seguiram para as etapas subsequentes de pré-processamento foram conforme Tabela 1.

Tabela 1 – Quantidades de imagens selecionadas para o treinamento da rede neural convolucional, separadas por grupo.

| **expressão** | **Imagens** |
|---|---|
| Neutra | 3957 |
| Raiva | 2870 |
| Felicidade | 2792 |
| Nojo | 2369 |
| Medo | 2055 |
| Tristeza | 2236 |
| Surpresa | 2786 |
| **Total** | **19065** |

Fonte: Elaborado pelo autor

Após a seleção das imagens de treinamento, um processo de recorte da Região de Interesse (ROI) foi aplicado a cada amostra. O objetivo dessa etapa é simplesmente a eliminação de informações de fundo, presentes nas capturas para evitar processamento desnecessário ou até mesmo reconhecimento de padrões não relevantes por parte da CNN. A identificação da ROI foi executada por meio do método *Cascade Classifier*, proposto por Viola e Jones (2004), implementado na própria biblioteca *OpenCV*. Nessa mesma etapa, um redimensionamento foi aplicado nas imagens para que se pudesse padronizar o conjunto de dados. As imagens foram traduzidas para uma resolução de 180x180 aplicando o método de interpolação por área, também presente na mesma biblioteca.

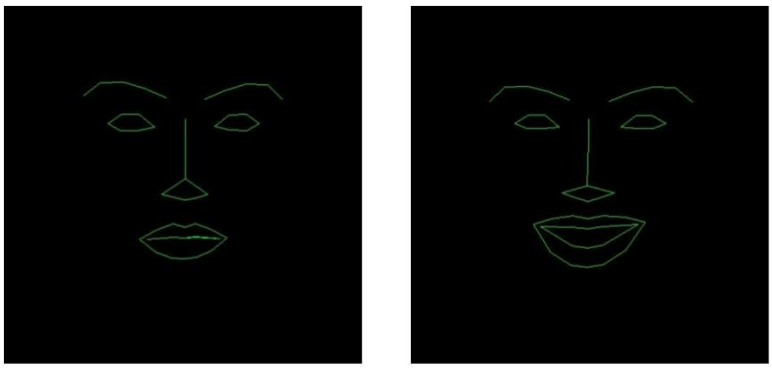
Ainda a partir da identificação do posicionamento do rosto, um alinhamento das imagens foi feito com o intuito de garantir que todas as faces estivessem em posição natural, ou seja, alinhadas horizontalmente em relação ao pontos centrais dos olhos. Essa tarefa foi executada com o auxílio do método *FaceAligner*, disponível no conjunto de ferramentas *face_utils* da biblioteca *imutils*.

Para evitar o reconhecimento de padrões sem relevância para o reconhecimento de expressões uma etapa adicional de extração de *landmarks* das amostras foi aplicada. Com esse processo, foram identificados 68 pontos relevantes da face com o auxílio da biblioteca

*dlib*. Dos 68 pontos encontrados, 8 deles (referentes ao contorno inferior do queixo) foram descartados, por serem considerados portadores de pouca ou nenhuma informação relevante para a classificação proposta por esse trabalho. Esta etapa do pré processamento auxilia na remoção de interferências externas que poderiam impactar nos resultados da rede, como luz ambiente, por exemplo, ou até mesmo características individuais do estudante, como cabelo ou cor da pele.

Por fim, uma etapa de espelhamento foi aplicada a todo o *dataset* resultante, com o intuito de ampliar a quantidade de amostras a serem apresentadas para a CNN possibilitando melhor aprendizado. Ao término de todas as etapas de pré-processamento, o conjunto de dados foi composto por 38130 imagens conforme o padrão apresentado na Figura 10.

Figura 10 – Resultado do pré-processamento do conjunto de dados a ser submetido a CNN. A esquerda pode-se observar uma amostra representante da expressão neutra, enquanto a imagem da direita representa o pico da expressão de felicidade.



Fonte: Elaborado pelo autor

### 5.3.2 Treinamento

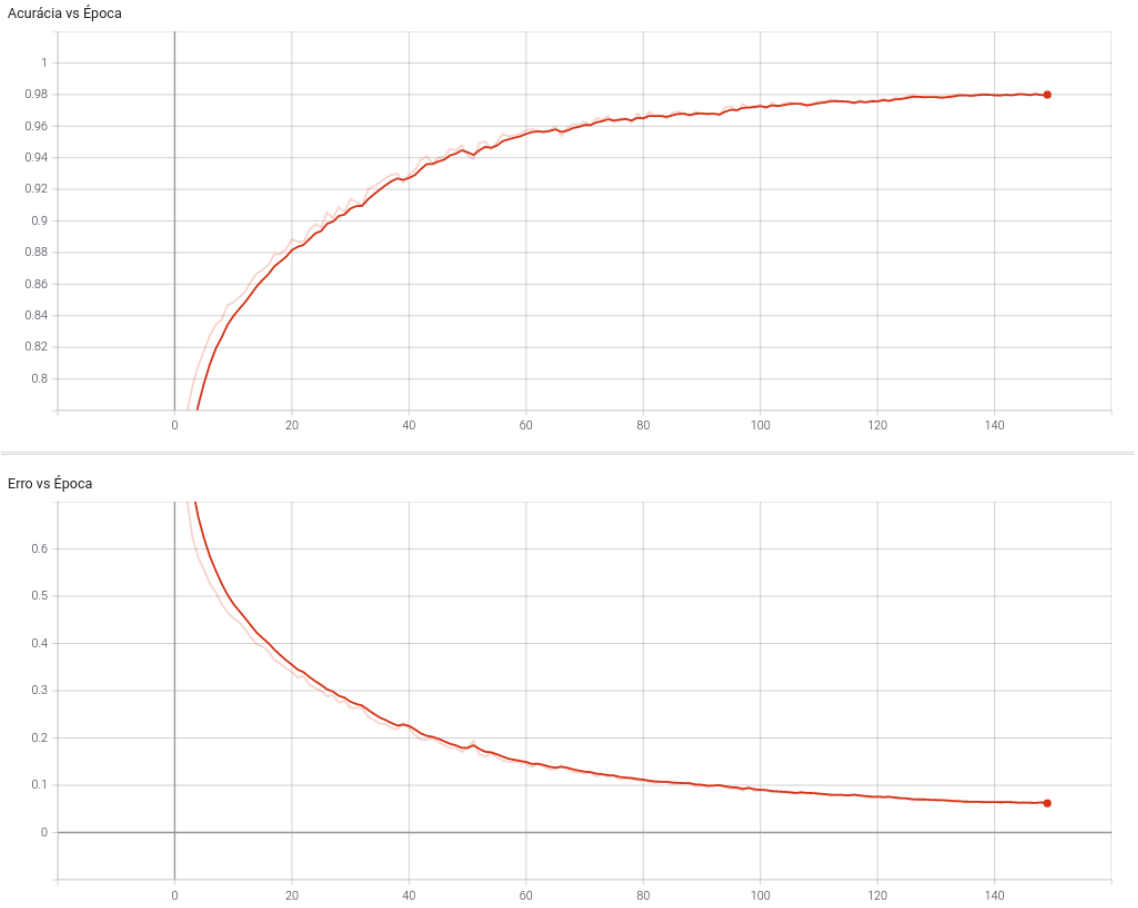
Essa etapa do processo de desenvolvimento do modelo proposto é onde ocorrem as iterações sobre o conjunto de dados, pré-processado, com o intuito de minimização da função de erro, um dos principais componentes de avaliação do modelo. As amostras preparadas anteriormente são inseridas na rede iniciando o processo de aprendizado e regendo ajustes nos pesos do modelo até a obtenção da acurácia desejada.

É importante salientar que, para que a rede possa ser avaliada de forma justa e se obtenha o real valor de acurácia alcançado durante o processo de treinamento, apenas uma parcela do conjunto de dados é apresentada para a CNN nessa etapa. Sendo assim, o treinamento foi conduzido com 70% do conjunto total resultante do pré-processamento (26691 imagens), 15% das imagens (5720 amostras) foram utilizadas para validação de treinamento e 15% foram reservadas para testes posteriores.

A separação das imagens em grupos de treinamento e validação se deu de forma aleatória, onde os subconjuntos de cada expressão foram embaralhados e divididos em parcelas para cada uma das etapas, conforme as proporções apresentadas acima.

Este processo, por ser o elemento que envolve o maior custo computacional de

Figura 11 – Representação da curva de aprendizado da rede na qual são apresentados a acurácia do modelo ao longo das épocas de treinamento no gráfico superior e o erro em função das mesmas épocas no gráfico inferior. O valor de acurácia é medido em relação a quantidade de amostras categorizadas pela rede de forma correta e, por outro lado, o valor de erro é referente a porcentagem atribuída a expressões diferentes daquela considerada correta. Pode-se perceber um grande aumento no nível de acerto da rede nas primeiras épocas e, a medida que o tempo passa, um ponto de convergência é alcançado.




Fonte: Elaborado pelo autor

todo o desenvolvimento, foi executado durante 1,5 horas, até que alcançou-se um ponto de convergência na medida de acurácia da rede, após 150 épocas de treinamento. O processo de aprendizado da rede pode ser observado na Figura 11.

## 5.4 Implantação no STI MAZK

O modelo de CNN apresentado neste trabalho foi desenvolvido utilizando a linguagem de programação Python por conta, principalmente, das bibliotecas disponíveis para tal. Para a viabilização da integração com a ferramenta de tutoria inteligente, foi necessária a criação de um micro serviço com o *framework* Flask, que possibilita o recebimento de requisições *http* de forma isolada à aplicação principal do MAZK. Além da criação do

micro serviço responsável pela classificação das imagens, foi necessária a implementação da captura das imagens no STI, bem como o *feedback* a ser fornecido ao tutor humano durante a interação.

Dentro do MAZK, ao ingressar em uma sala onde é efetuado o reconhecimento de expressões, é solicitado ao aluno a liberação do uso de sua câmera para dar início ao processo. Caso o estudante não deseje habilitar esse recurso, sua imagem não é capturada e o fluxo de aprendizado é continuado de forma habitual. Por outro lado, se o aluno aceitar participar do processo de reconhecimento das expressões e habilitar a câmera do seu dispositivo, o sistema imediatamente inicia a captura de imagens da sua face de forma temporal e as submete ao micro serviço citado acima. O micro serviço, ao receber uma requisição do sistema principal, acompanhada de uma imagem da face, submete essa imagem a todas as etapas de pre-processamento mencionadas na subseção 5.3.1 e, posteriormente, avalia a expressão aplicando-a no modelo previamente treinado. A partir disso, o resultado da classificação é retornado à aplicação principal para que essa informação seja repassada ao tutor humano que acompanha a atividade de forma síncrona.

A resposta retornada pelo micro serviço, com os dados de classificação, consiste em um grupo de informações no formato JSON, no qual são compreendidos os rótulos das emoções e as probabilidades atribuídas pelo modelo a cada uma delas, conforme exemplo da Figura 12.

Figura 12 – Resposta gerada pelo micro serviço de classificação. Pode-se observar um campo nomeado *success* que, nesse exemplo, possui o valor verdadeiro (*true*). Caso ocorra algum problema no processo de classificação da imagem, como a não detecção de nenhuma face, por exemplo, esse campo é retornado como falso (*false*) e o as classificações como uma lista vazia([]) e, sendo assim, o tutor ignora tal amostra para para geração do *feedback* ao professor.

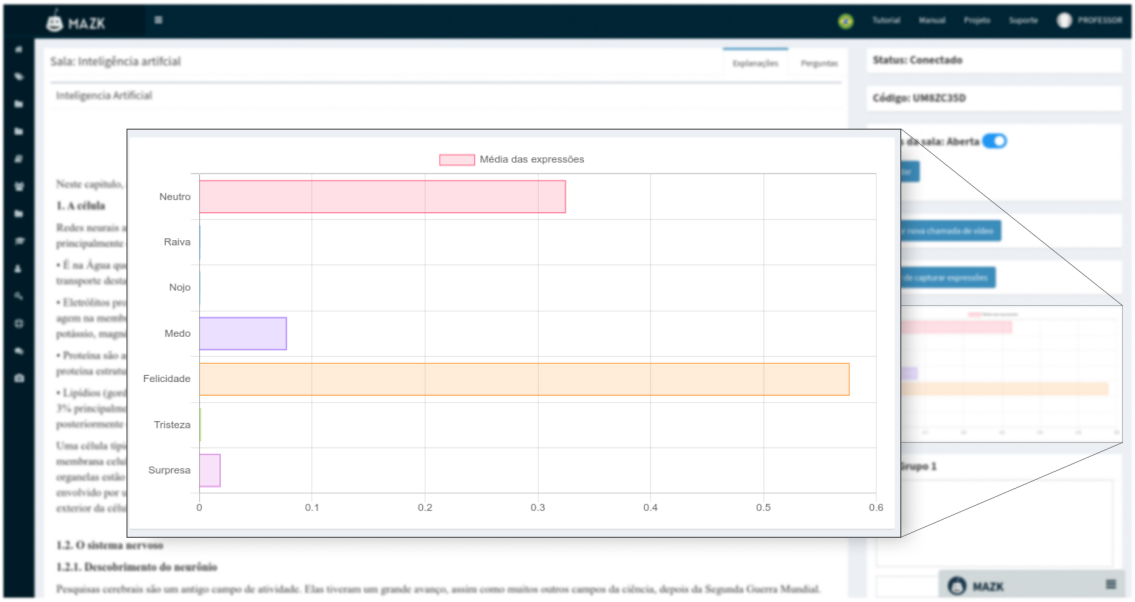

```
{
    "labels_set": [
        "Neutral",
        "Anger",
        "Disgust",
        "Fear",
        "Happy",
        "Sadness",
        "Surprise"
    ],
    "score": [
        6.682493676635204e-06,
        0.007353377062827349,
        0.9917601943016052,
        1.786614802767872e-07,
        1.8590614558888774e-07,
        0.0008788497652858496,
        3.786045681408723e-07
    ],
    "success": true
}
```


Fonte: Elaborado pelo autor

Conforme a resposta recebida da etapa de classificação, uma média de cada expressão é calculada levando-se em conta as amostras de todos os alunos participantes da

captura no momento em questão e, a partir disso, é gerado um gráfico conforme Figura 13, para servir como um *feedback* visual ao professor.

Figura 13 – A média de cada uma das emoções apresentadas no gráfico é efetuada a partir da predição da CNN para cada imagem submetida a ela, ou seja, o valor de cada expressão é calculado separadamente, levando em conta a proporção da sua presença em cada amostra de acordo com a classificação do modelo.




Fonte: (MAZK, 2021)

# 6 Resultados Experimentais

## 6.1 Avaliação do Modelo

Ao final da etapa de treinamento, detalhada na subseção 5.3.2, o modelo proposto foi capaz de classificar corretamente $\approx 98\%$ das amostras a ele apresentadas, ou seja, a acurácia alcançada pela rede neural convolucional foi de 0,9801 como pode ser visualizado na Figura 9. No entanto, a acurácia obtida pelo modelo muitas vezes não representa o alcance do objetivo geral de sua aplicação e, por conta disso, é importante a utilização de outras métricas. Conforme exposto por Goodfellow, Bengio e Courville (2016), levando-se em consideração a aplicação de um classificador que detecta a presença de um evento raro, como uma doença que se manifesta em uma a cada um milhão de pessoas. Nesse caso, podemos atingir facilmente uma acurácia de 99,9999% simplesmente programando o classificador para que sempre classifique as imagens como falso em termos de presença de tal doença. Para contornar esse problema, os próprios autores sugerem a aplicação das métricas de *precision* e *recall*, onde *precision* representa as predições feitas pelo modelo de forma correta e *recall*, por outro lado, representa a fração de eventos verdadeiros detectados. Com a obtenção de um modelo ideal, busca-se alcançar valores próximos a 1 para ambos *precision* e *recall* e, para podermos avaliar de forma mais justa a correlação entre esses

dois métodos, a métrica *f1-score* é implementada da seguinte forma:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{2}$$

Assim, caso os valores de *precision* e *recall* sejam muito diferentes um do outro, o valor dessa métrica será baixo e, em um caso ideal, alcançar-se-á *F1-score* = 1.

Os resultados obtidos para todas as métricas mencionadas acima estão presentes na Tabela 2.

Tabela 2 – Apresentação das três métricas utilizadas para a validação do modelo proposto para cada uma das expressões consideradas e, ao final, a média ponderada de cada uma das métricas levando em conta todas as emoções.

| **Expressão** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| **Neutro** | 0,95 | 0,96 | 0,95 |
| **Raiva** | 0,97 | 0,98 | 0,98 |
| **Nojo** | 0,99 | 0,98 | 0,98 |
| **Medo** | 0,97 | 0,97 | 0,97 |
| **Felicidade** | 0,99 | 0,98 | 0,98 |
| **Tristeza** | 0,99 | 0,98 | 0,99 |
| **Surpresa** | 0,98 | 0,98 | 0,98 |
| **Média ponderada** | **0,98** | **0,98** | **0,98** |

Fonte: Elaborado pelo autor

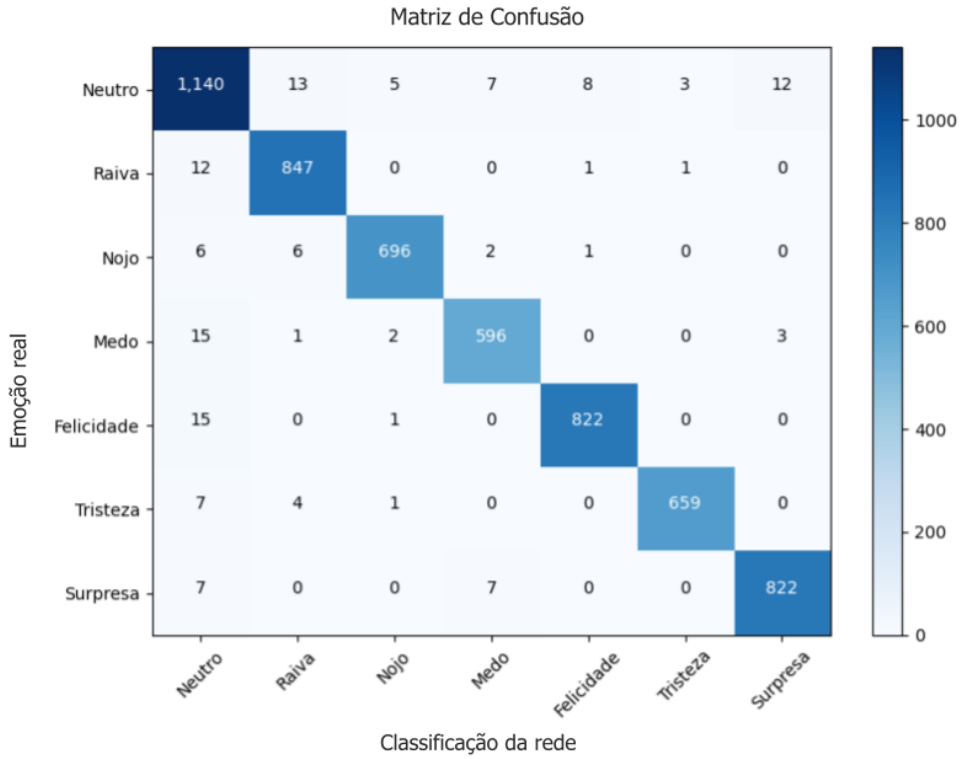
A avaliação da rede em termos dessas métricas pode ser visualizada também na matriz de confusão (Figura 14), bem como a correlação de classificações entre as emoções consideradas.

## 6.2 Avaliação da Implantação no STI MAZK

Para que fosse possível aplicar o modelo de classificação no STI MAZK, foi preciso garantir que os recursos computacionais necessários para sua aplicação não excedessem o poder computacional disponível no ambiente de produção. Para isso, foram verificadas as utilizações dos principais recursos durante a execução da ferramenta: CPU e Memória RAM. Através do acompanhamento do servidor e de observações dos índices de utilização de memória, pode-se perceber que o uso de RAM pelo micro serviço aplicado é constante. Isso ocorre pelo fato de que o serviço funciona a partir do carregamento do modelo na RAM, e disso em diante, não necessita de recurso adicional referente a esse parâmetro. Contudo, a utilização de memória, durante os testes preliminares, atingiu um valor de $\approx 7\%$ do total disponível no servidor, o que corresponde a aproximadamente 420MB. No momento do teste aplicado, com a adição desse valor ao uso total de memória, o servidor chegou a atingir uma utilização de $\approx 62,5\%$ do total de RAM disponível, o que leva a acreditar que tal recurso não é um problema para a aplicação do modelo de classificação.

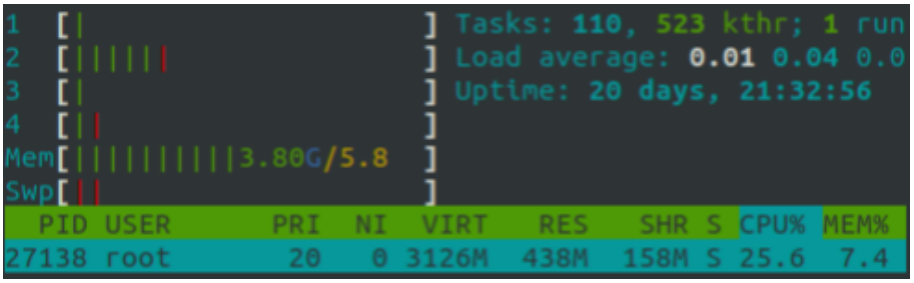
Por outro lado, na utilização de recursos de processador (CPU) pode-se perceber um pico durante a avaliação de uma amostra, onde o algoritmo atinge $\approx 25\%$ de utilização de um núcleo de processamento do servidor, conforme pode ser observado na Figura 15. Contudo, pode-se também observar que o uso do recurso citado não inviabiliza a aplicação

Figura 14 – Matriz de confusão onde podem ser observados os principais pontos de acerto e falha nas classificações da rede. A matriz foi gerada a partir da classificação de 15% do conjunto de dados, sendo que essas amostras nunca haviam sido apresentadas ao modelo anteriormente.




Fonte: Elaborado pelo autor

Figura 15 – Utilização de CPU e memória pelo micro serviço de classificação de emoções. Os três valores que constam após o rótulo *Load average* (0.01, 0.04 e 0.0) correspondem ao uso total de CPU nos últimos 1, 5 e 15 minutos, respectivamente.




Fonte: Elaborado pelo autor

do modelo pois, mesmo utilizando uma quantidade considerável do poder de processamento

disponível, a aplicação principal ainda possui a sua disposição cerca de 87% de CPU, sendo que sua utilização média sequer atinge 1% do total durante seu uso normal.

# 7 Conclusão e Trabalhos Futuros

O avanço da tecnologia, tanto em termos de *hardware* como dos algoritmos associados à visão computacional tem proporcionado progresso em diversas áreas com a aplicação de inteligência artificial e, além disso, possibilitado melhoramento de soluções existentes. Neste trabalho foi apresentado um modelo de rede neural convolucional que possibilita a detecção de expressões presentes em imagens da face, bem como a aplicação dessa solução no sistema tutor inteligente MAZK, para auxílio no processo de ensino-aprendizagem.

Em termos de métricas de desempenho, um resultado bem expressivo foi obtido para a classificação das seguintes expressões: neutro, raiva, nojo, medo, felicidade, tristeza e surpresa, atingindo uma média ponderada dentre as expressões de $\approx 98\%$. Além disso, o método foi aplicado no Sistema Tutor Inteligente MAZK, onde foi possível observar que a solução proposta se destacou em relação aos casos de aplicações observados na Seção 4 sem sacrificar o desempenho do sistema principal.

As possibilidades de utilização do modelo construído são inúmeras e se mantém em aberto, tendo em vista que ele foi implantado como um micro serviço no servidor do STI, podendo ser utilizado de qualquer parte do sistema, não somente nas salas, e até mesmo servir como uma ferramenta isolada ao sistema MAZK. Entretanto, algumas aplicações para o próprio tutor podem ser citadas. Uma delas é a integração da classificação de expressões ao modelo do aluno, conceito de inteligência artificial aplicado no MAZK. Essa aplicação possibilita à ferramenta a interpretação dos dados para utilização como base na tomada de decisão, aperfeiçoando o *loop* interno do tutor, por exemplo, através de um *feedback* visual, deixando a interação com o estudante mais robusta.

Além disso, o algoritmo deve ser testado com mais indivíduos e diferentes perfis de usuários dentro das ferramentas do MAZK, podendo ser integrado a outras técnicas de IA para obtenção de métricas relevantes para o processo de ensino-aprendizagem. De toda forma, uma análise qualitativa do método proposto deve ser conduzida em um trabalho futuro, mediante sua aplicação em situações reais de utilização do MAZK. Nessa aplicações poderá ser analisada a aceitação da funcionalidade por parte dos professores e alunos por meio de *feedbacks* solicitados aos mesmos.

Em resumo, o modelo apresentado neste trabalho foi capaz de alcançar elevados índices de desempenho, tanto em sua precisão, quanto no custo computacional envolvido na sua execução. Ao ser incorporado no modelo do aluno, o método desenvolvido pode fornecer base de informações ao STI, promovendo uma melhor tomada de decisão durante a interação do aluno com o tutor e, sendo assim, pode agregar valor à ferramenta MAZK por meio da sua contribuição nesse processo.

# Referências


ABADI, Martın *et al.* Tensorflow: A system for large-scale machine learning. In: 12TH {USENIX} symposium on operating systems design and implementation ({OSDI} 16). [S.l.: s.n.], 2016. P. 265–283.

AIFANTI, Niki; PAPACHRISTOU, Christos; DELOPOULOS, Anastasios. The MUG facial expression database. In: IEEE. 11TH International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10. [S.l.: s.n.], 2010. P. 1–4.

AKPUTU, Kingsley Oryina; SENG, Kah Phooi; LEE, Yun Li. Facial emotion recognition for intelligent tutoring environment. In: 2ND International Conference on Machine Learning and Computer Science (IMLCS'2013). [S.l.: s.n.], 2013. P. 9–13.

ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: IEEE. 2017 International Conference on Engineering and Technology (ICET). [S.l.: s.n.], 2017. P. 1–6.

ALI, Hasimah *et al.* Facial emotion recognition using empirical mode decomposition. **Expert Systems with Applications**, Elsevier, v. 42, n. 3, p. 1261–1277, 2015.

ALZANTOT, Moustafa *et al.* Rstensorflow: Gpu enabled tensorflow for deep learning on commodity android devices. In: PROCEEDINGS of the 1st International Workshop on Deep Learning for Mobile Systems and Applications. [S.l.: s.n.], 2017. P. 7–12.

AMMAR, Mohamed Ben *et al.* The affective tutoring system. **Expert Systems with Applications**, Elsevier, v. 37, n. 4, p. 3013–3023, 2010.

BALDASSARRI, Sandra *et al.* Affective-aware tutoring platform for interactive digital television. **Multimedia Tools and Applications**, Springer, v. 74, n. 9, p. 3183–3206, 2015.

BARRÓN ESTRADA, Marıa Lucıa; ZATARAIN CABADA, Ramón; HERNÁNDEZ PÉREZ, Yasmın. Tutor inteligente con reconocimiento y manejo de emociones para Matemáticas. **Revista electrónica de investigación educativa**, Universidad Autónoma de Baja California, Instituto de Investigación y ..., v. 16, n. 3, p. 88–102, 2014.

BISWAS, Suparna; SIL, Jaya. An efficient expression recognition method using contourlet transform. In: PROCEEDINGS of the 2nd International Conference on Perception and Machine Intelligence. [S.l.: s.n.], 2015. P. 167–174.

BITTENCOURT, William Nunes *et al.* A utilização do tutor inteligente MAZK no processo de ensino-aprendizagem, 2018.

EKMAN, Paul; FRIESEN, Wallace V. Constants across cultures in the face and emotion. **Journal of personality and social psychology**, American Psychological Association, v. 17, n. 2, p. 124, 1971.

FRIESEN, E; EKMAN, Paul. Facial action coding system: a technique for the measurement of facial movement. **Palo Alto**, Consulting Psychologists Press, v. 3, 1978.

GALAFASSI, Cristiano *et al.* EvoLogic: Sistema Tutor Inteligente para Ensino de Lógica. In: SBC. ANAIS do XLVII Seminário Integrado de Software e Hardware. [S.l.: s.n.], 2020. P. 222–233.

GAN, Yijun. Facial Expression Recognition Using Convolutional Neural Network. In: ACM. PROCEEDINGS of the 2nd International Conference on Vision, Image and Signal Processing. [S.l.: s.n.], 2018. P. 29.

GOLDSBOROUGH, Peter. A tour of tensorflow. **arXiv preprint arXiv:1610.01178**, 2016.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GRAESSER, Arthur C; CONLEY, Mark W; OLNEY, Andrew. Intelligent tutoring systems. **APA educational psychology handbook, Vol 3: Application to learning and teaching.**, American Psychological Association, p. 451–473, 2012.

GUO, Yanming *et al.* Deep learning for visual understanding: A review. **Neurocomputing**, Elsevier, v. 187, p. 27–48, 2016.

HAENSCH, Wilfried; GOKMEN, Tayfun; PURI, Ruchir. The next generation of deep learning hardware: Analog computing. **Proceedings of the IEEE**, IEEE, v. 107, n. 1, p. 108–122, 2018.

HAPPY, SL; ROUTRAY, Aurobinda. Automatic facial expression recognition using features of salient facial patches. **IEEE transactions on Affective Computing**, IEEE, v. 6, n. 1, p. 1–12, 2014.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer Science & Business Media, 2009.

HU, Min *et al.* Video facial emotion recognition based on local enhanced motion history image and CNN-CTSLSTM networks. **Journal of Visual Communication and Image Representation**, Elsevier, v. 59, p. 176–185, 2019.

HWANG, Gwo-Jen. A conceptual map model for developing intelligent tutoring systems. **Computers & Education**, Elsevier, v. 40, n. 3, p. 217–235, 2003.

JOSUÉ, Mupenza Mupenza *et al.* O ensino e estudo de Inteligência Artificial num paıs francófono utilizando o sistema tutor inteligente MAZK. Araranguá, SC, 2018.

KANADE, Takeo; COHN, Jeffrey F; TIAN, Yingli. Comprehensive database for facial expression analysis. In: IEEE. PROCEEDINGS Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580). [S.l.: s.n.], 2000. P. 46–53.

LANGNER, Oliver *et al.* Presentation and validation of the Radboud Faces Database. **Cognition and emotion**, Taylor & Francis, v. 24, n. 8, p. 1377–1388, 2010.

LI, Qing *et al.* Medical image classification with convolutional neural network. In: IEEE. 2014 13th international conference on control automation robotics & vision (ICARCV). [S.l.: s.n.], 2014. P. 844–848.

LIN, Hao-Chiang Koong *et al.* Employing Textual and Facial Emotion Recognition to Design an Affective Tutoring System. **Turkish Online Journal of Educational Technology-TOJET**, ERIC, v. 11, n. 4, p. 418–426, 2012.

LOPES, André Teixeira *et al.* Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. **Pattern Recognition**, Elsevier, v. 61, p. 610–628, 2017.

LUCEY, Patrick *et al.* The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: IEEE. 2010 ieee computer society conference on computer vision and pattern recognition-workshops. [S.l.: s.n.], 2010. P. 94–101.

MALDANER, Natalia *et al.* Computação afetiva aplicada à educação: uma proposta ao Sistema Tutor Inteligente MAZK. Araranguá, SC, 2019.

MAZK. **Sistema Tutor Inteligente MAZK**. [S.l.: s.n.], 2021. Acessado em 17.04.2021. Disponível em: <<https://www.mazk.labtec.ufsc.br>>.

MOHRI, Mehryar; ROSTAMIZADEH, Afshin; TALWALKAR, Ameet. **Foundations of machine learning**. [S.l.]: MIT press, 2018.

MOHSENI, Sina; ZAREI, Niloofar; RAMAZANI, Saba. Facial expression recognition using anatomy based facial graph. In: IEEE. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). [S.l.: s.n.], 2014. P. 3715–3719.

MORO, Francielli Freitas *et al.* Protótipo de um chatbot para auxiliar o professor na utilização de um sistema tutor inteligente Mazk, 2019.

MURPHY, Kevin P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012.

POZZEBON, Eliane *et al.* Um modelo para suporte ao aprendizado em grupo em sistemas tutores inteligentes. Florianópolis, SC, 2008.

PUTHANIDAM, Roshni Velluva; MOH, Teng-Sheng. A Hybrid Approach for Facial Expression Recognition. In: ACM. PROCEEDINGS of the 12th International Conference on Ubiquitous Information Management and Communication. [S.l.: s.n.], 2018. P. 60.

RAWAT, Waseem; WANG, Zenghui. Deep convolutional neural networks for image classification: A comprehensive review. **Neural computation**, MIT Press, v. 29, n. 9, p. 2352–2449, 2017.

QI-RONG, Chen. Research on intelligent tutoring system based on affective model. In: IEEE. 2010 Second International Conference on Multimedia and Information Technology. [S.l.: s.n.], 2010. v. 1, p. 7–9.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014.

SILVA, Viviane Izabel da *et al.* Um modelo para a utilização da metodologia ativa aprendizagem baseada em casos no sistema tutor inteligente Mazk, 2019.

SINGH, Sinam Ajitkumar; MEITEI, Takhellambam Gautam; MAJUMDER, Swanirbhar. Short PCG classification based on deep learning. In: DEEP Learning Techniques for Biomedical and Health Informatics. [S.l.]: Elsevier, 2020. P. 141–164.

SLIMANI, K *et al.* Facial emotion recognition: A comparative analysis using 22 LBP variants. In: ACM. PROCEEDINGS of the 2nd Mediterranean Conference on Pattern Recognition and Artificial Intelligence. [S.l.: s.n.], 2018. P. 88–94.

TURABZADEH, Saeed *et al.* Real-time emotional state detection from facial expression on embedded devices. In: IEEE. 2017 Seventh International Conference on Innovative Computing Technology (INTECH). [S.l.: s.n.], 2017. P. 46–51.

VANLEHN, Kurt. The behavior of tutoring systems. **International journal of artificial intelligence in education**, IOS Press, v. 16, n. 3, p. 227–265, 2006.

VIDOTTO, Kajiana Nuernberg Sartor *et al.* Ambiente Inteligente de Aprendizagem MAZK com alunos do Ensino Fundamental II na disciplina de Ciências. In: 1. BRAZILIAN Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). [S.l.: s.n.], 2017. v. 28, p. 1367.

VIOLA, Paul; JONES, Michael J. Robust real-time face detection. **International journal of computer vision**, Springer, v. 57, n. 2, p. 137–154, 2004.

WANG, Jichen; LIN, Jun; WANG, Zhongfeng. Efficient convolution architectures for convolutional neural network. In: IEEE. 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP). [S.l.: s.n.], 2016. P. 1–5.

WU, Yanwen; LIU, Wei; WANG, Jianbo. Application of emotional recognition in intelligent tutoring system. In: IEEE. FIRST International Workshop on Knowledge Discovery and Data Mining (WKDD 2008). [S.l.: s.n.], 2008. P. 449–452.

YAMASHITA, Rikiya *et al.* Convolutional neural networks: an overview and application in radiology. **Insights into imaging**, Springer, v. 9, n. 4, p. 611–629, 2018.

ZATARAIN-CABADA, Ramón *et al.* Java Tutoring System with Facial and Text Emotion Recognition. **Res. Comput. Sci.**, v. 106, p. 49–58, 2015.