

Reconhecimento Facial Utilizando Redes Neurais Convolucionais Para Auxiliar na Automatização de Processos no Campus Santo Antônio de Jesus do IFBA

Aluno: Daniel Almeida Caitano

O que é Inteligência Artificial?

A Inteligência Artificial (IA) é a simulação de processos de inteligência humana por máquinas, especialmente sistemas de computador.

O Aprendizado de Máquina (machine learning): máquinas executam seus trabalhos por conta própria.

A Aprendizagem Profunda (Deep Learning): Permite que computadores, através de uma rede neural, resolvam problemas como reconhecimento de fala e imagem.

Aplicações da Inteligência Artificial

Inteligência artificial no reconhecimento facial no facebook;

Indicação de produtos e conteúdos através de inteligência artificial;

Chatbots de contato com o cliente através de inteligência artificial;

Assistentes por voz de inteligência artificial;

Sistema de pesquisa do google.

Motivação

Auxiliar o desenvolvimento de futuros processos envolvendo IA, no Instituto Federal da Bahia (IFBA) campus Santo Antônio de Jesus, e com isso automatizar processos na universidade.

Problema

Quais as contribuições que a aplicação de redes neurais convolucionais poderia trazer para o campus Santo Antônio de Jesus do IFBA?

Graças a evolução tecnológica, com ajuda das redes neurais convolucionais, pode ser criado um sistema de reconhecimento facial que automatizaria e facilitaria tarefas no campus.

Controle de entrada e saída no campus;

Sistema de presença automática;

Identificação dos alunos para conceder acessos e/ou permissões.

Metodologia

Este trabalho utilizou uma abordagem de processamento de imagens e reconhecimento de padrões para detectar e conseguir fazer a classificação das imagens.

Algumas linguagens, através das suas bibliotecas, podem ser utilizadas para o desenvolvimento da rede neural como: python, java, Javascript, Swift.

Alguns modelos de rede populares são: ImageNet Competition, AlexNet, VGG, Inception/GoogleNet, Resnet.

Metodologia

ML5.js: Construído em cima do tensorflow.js que possibilita o uso Machine Learning na web com o processamento de dados através da GPU.

Tensorflow.js: É uma biblioteca criada pelo Google para trabalhar Machine Learning no JavaScript, foi construído baseado no TensorFlow biblioteca escrita em Python.

P5.js: Possui um conjunto de funcionalidades para trabalhar com objetos HTML5 para texto, vídeo, imagem, webcam e som, além de desenhos com o canvas.

Desenvolvimento

Dataset: foi construído com conjuntos de dados de pessoas famosas do site Kaggle. Para este estudo foi usado imagens de 30 pessoas e cada pessoa contém cerca de 33 imagens.

Desenvolvimento



Ator norte-americano Chris Evans



Apresentadora norte-americana Oprah Winfrey

Desenvolvimento



Cantora norte-americana Rihanna



Política alemã Angela Merkel

Desenvolvimento

A arquitetura do sistema segue a estrutura típica de um processamento digital de imagens com sistema de reconhecimento de padrões utilizando o ml5.js e consiste em cinco etapas:

Desenvolvimento

Etapas 1: Carregar dados ou criar alguns dados (data augmentation):

25 imagens de cada pessoa foram carregadas com ajuda do p5.js com o método de carregar imagem(loadImage) que permite que as fotos sejam carregadas de um caminho e se crie um objeto Image a partir dela.

Desenvolvimento

Etapas 2: Definir as opções de rede neural e inicializá-la

A rede foi inicializada utilizando o método de rede neural do ml5.js (neuralNetwork). Alguns parâmetros precisaram ser passados:

Classificação de imagem (imageClassification);

Input no qual especificamos a largura(width) e altura(height) das imagens e seus canais (channels). As imagens foram redimensionadas com o método do p5.js (resize);

A rede foi configurada com base na arquitetura resnet.

Desenvolvimento

Model Summary			
Layer Name	Output Shape	# Of Params	Trainable
conv2d_Conv2D1	[batch,48,48,64]	12,608	true
batch_normalization_BatchNormalization1	[batch,48,48,64]	256	true
activation_Activation1	[batch,48,48,64]	0	true
max_pooling2d_MaxPooling2D1	[batch,24,24,64]	0	true
conv2d_Conv2D2	[batch,24,24,64]	36,928	true
batch_normalization_BatchNormalization2	[batch,24,24,64]	256	true
activation_Activation2	[batch,24,24,64]	0	true
conv2d_Conv2D3	[batch,24,24,64]	36,928	true
batch_normalization_BatchNormalization3	[batch,24,24,64]	256	true
activation_Activation3	[batch,24,24,64]	0	true
conv2d_Conv2D4	[batch,24,24,64]	36,928	true
batch_normalization_BatchNormalization4	[batch,24,24,64]	256	true
activation_Activation4	[batch,24,24,64]	0	true
conv2d_Conv2D5	[batch,24,24,64]	36,928	true
batch_normalization_BatchNormalization5	[batch,24,24,64]	256	true

conv2d_Conv2D6	[batch,24,24,256]	295,168
batch_normalization_BatchNormalization6	[batch,24,24,256]	1,024
activation_Activation6	[batch,24,24,256]	0
conv2d_Conv2D7	[batch,24,24,256]	590,080
batch_normalization_BatchNormalization7	[batch,24,24,256]	1,024
activation_Activation7	[batch,24,24,256]	0
conv2d_Conv2D8	[batch,24,24,512]	1,180,160
batch_normalization_BatchNormalization8	[batch,24,24,512]	2,048
activation_Activation8	[batch,24,24,512]	0
conv2d_Conv2D9	[batch,24,24,512]	2,359,808
batch_normalization_BatchNormalization9	[batch,24,24,512]	2,048
activation_Activation9	[batch,24,24,512]	0
global_average_pooling2d_GlobalAveragePooling2D1	[batch,512]	0
dense_Dense1	[batch,30]	15,390

Resumo do modelo gerado pelo ml5.js

Desenvolvimento

Etapas 3: Normalizar dados à rede neural

Nesta etapa adicionamos os dados para a rede neural e foi utilizado o método de adicionar dados (`addData`) do `ml5.js`.

Após isso, os dados precisaram ser normalizados. Para fazer isto foi utilizado o método `normalizeData` do `ml5.js`, assim os pixels das imagens foram convertidos para valores entre 0 e 1.

Desenvolvimento

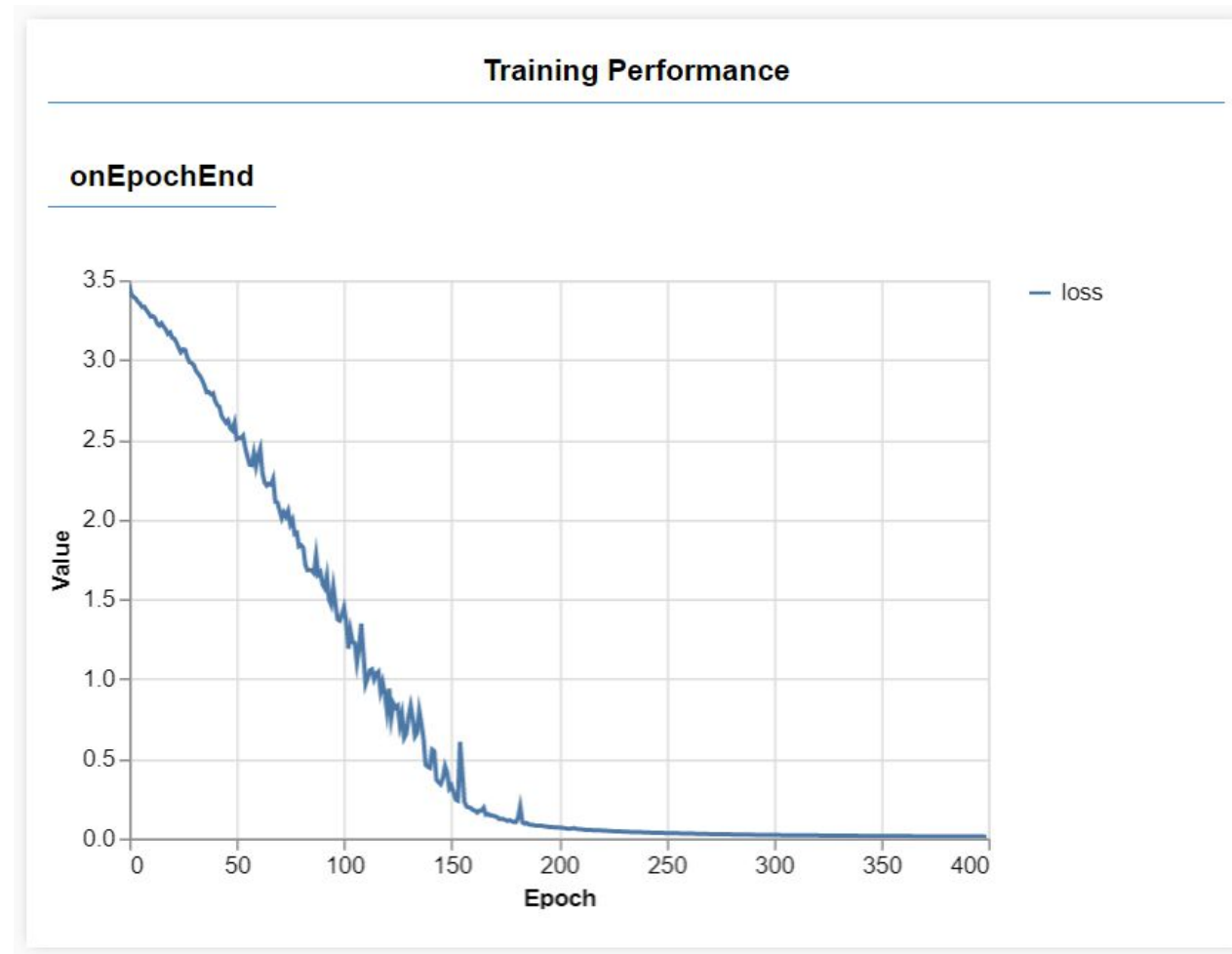
Etapa 4: Treinar a rede neural

A rede neural convolucional teve seu treinamento utilizando o método train do ml5.js.

O parâmetro epochs teve de ser usado para saber o número de épocas de treinamento, o número de 400 épocas foi escolhido.

o tamanho do batch (batch size) foi utilizado o número 75.

Desenvolvimento



Performance do treinamento gerado pelo ml5.js

Resultados

Etapa 5: Usar o modelo treinado para fazer uma classificação.

As imagens que foram usadas para treinar a rede foram usadas como input para verificar como a rede as classificava. Após os testes a rede retornou uma média de confiança de aproximadamente 99%.

Em seguida foram usadas 8 imagens, que não foram usadas para fazer o treinamento da rede neural, de cada pessoa para avaliar a classificação da rede, desta vez a rede retornou uma média de confiança de 72,37%.

Conclusão

Graças a média de confiança de 72,37%, é possível dizer que o projeto já apresenta grau inicial de desenvolvimento que permite a realização de testes com estudantes.

Com o resultado, espera-se que a rede possa ser aprimorada e utilizada para automatizar algumas tarefas no IFBA, a fim de melhorar e, facilitar as atividades no campus.

Com ela, seria possível obter identificação dos alunos para conceder acessos e/ou permissões, controle de entrada e saída no campus entre outras utilidades.

Referências

BURNS, Ed et al. What is artificial intelligence (AI)? Massachusetts: TechTarget Business Technology, 2022.

RUSK, Nicole. METHODS TO WATCH | SPECIAL FEATURE. vol. 13, no.1. Nature America, Inc., 2016.

SILVA FILHO, Darlan de Castro. Reconhecimento de Caracteres Utilizando Redes Neurais Convolucionais para Auxiliar nas Correções do Sistema Multiprova. Natal: Epidemiol. Uni-versidade Federal do Rio Grande do Norte, 2022.

FERREIRA, Michael Henrique Souza. Reconhecimento Facial Para Detecção De Emoções Utilizando Redes Neurais Convolucionais Com Tensorflow. Goiânia: Universidade Católica de Goiás, 2021.

DAMIÃO, Pedro Henrique Ferreira da Costa et al. RECOGSYS - Sistema de Contro-le de Acesso Utilizando Reconhecimento Facial. Juiz de Fora: IF Sudeste MG, 2021.

COSTA, Lucas Z. et al. Reconhecimento de Pessoas com e sem Máscara Usando Redes Neurais Convolucionais. Lages Instituto Federal de Santa Catarina, 2022.