Banco de Dados do IdentIF – Ponto Identificação Facial

Vinicius Barbosa Ribeiro[1], Eduardo Hiroshi Nakamura[1], Edson da Silva Castro[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Mato Grosso do Sul – Três Lagoas - MS

vinicius.ribeiro4@estudante.ifms.edu.br, eduardo.nakamura@ifms.edu.br, edson.castro@ifms.edu.br

Resumo

Este projeto se refere ao banco de dados do sistema IndentIF, que é utilizada de Visão Computacional para o Reconhecimento Facial dos funcionários do Instituto Federal do Mato Grosso do Sul e o registro de sua presença. Para distinguir cada pessoa este banco armazena informações de detalhes da face, de tempo, de hash criptográfico, nome e siape. O Software desenvolvido em C++ com a biblioteca QT e irá através da técnica de Inteligência Artificial gerar informações que serão armazenadas no SGDB(Sistema Gerenciador de Banco de Dados) PostgreSQL para posteriormente ser enviado para o SUAP(Sistema Unificado de Administração Pública) através de REST(Representational State Transfer). Tanto o software IdentIF e quanto o SGDB do mesmo foram implementados e são executados sobre o hardware do NVIDIA Jetson, que roda Linux e possibilita o uso de aceleração do processamento através da GPU(Unidade de processamento gráfico).

Palavras-chave: C++, Inteligência Artificial, PostgreSQL.

Introdução

A Inteligencia Artificial é a área da Ciência da Computação que atualmente esta em maior destaque. Dentre uma de suas técnicas esta a Visão Computacional que permite que executemos o reconhecimento das características faciais de uma pessoa e consequentemente sua identificação. Os algoritmos de Redes Neurais são fundamentais nesse processo, pois são eles que aprendem como distinguir uma pessoa de outra através de imagens. Na programação o processamento de rosto e de neurônios é feito através operações com matrizes e utilizar um hardware com GPU agiliza o processo.

Redes Neurais precisam ser treinadas para poder diferenciar um rosto de outro e ter uma base de dados de fotos permite que elas aprendam como fazer isso. A identificação facial acontece através do reconhecimento de pontos específicos do rosto. Uma imagem só é reconhecida como exatamente igual quando comparada a ela mesma, assim também acontece com rostos, uma leve inclinação ou diferença de iluminação faz com que a comparação não seja exata, resultando em uma aproximação.

Uma Rede Neural bem treinada consegue identificar rostos com variações de ângulo e luminosidade através de aproximações que tentem a ser muito próximas a cem por cento, mas nunca o valor completo.

Os valores importantes para a identificação facial que a Rede Neural reconhece são os contornos dos olhos, nariz e boca e rosto.

Após fazer uma identificação o sistema ficara com um vetor de valores que sera armazenado no PostgreSQL(MILANI, 2008) juntamente com tempo da identificação, hash criptográfico, nome e siape. O armazenamento se torna necessário pois caso exista reclamação os dados estarão la para comparação devido a exatidão não chegar os cem por cento.

Além de ser um banco de dados robusto, o PostgreSQL facilita a busca do vetor e dos outros dados para o envio via método REST para o SUAP da Reitoria.

O REST(LECHETA, 2015) são métodos de programação de pagina de internet que facilitam e padronizam os dados trafegados, sendo esse também o método aceito pelo SUAP.

O Sistema Operacional Linux permitiu que o sistema fosse desenvolvido com a estabilidade necessária em um ecossistema de desenvolvimento, ademais por ser livre não adicionou custo ao projeto.

C++(SWAN, 2000) e a biblioteca QT permitiram a interface entre processamento, armazenamento e pesquisa dos dados.

Metodologia

Foram levantados os requisitos de hardware identificando que o NVIDIA Jetson possui o processamento necessário em CPU(Unidade Central de Processamento) e GPU para ser um terminal de ponto eletrônico.

Nos requisitos de software identificou-se que a biblioteca OpenCV(KAEHLER, 2017) feita em C++ possibilitaria o reconhecimento facial e utilizaria da aceleração da GPU, Em conciliação a linguagem a biblioteca QT permitiu a comunicação com o banco e a interação com o desenvolvimento do outro plano de trabalho.

Para a modelagem do banco de dados no PostgreSQL foram considerados os dados advindos da detecção e reconhecimento facial criando campos coerentes com eles. Adicionados de dados de tempo, segurança e identificação das pessoas(nome e siape).

Na Figura 1 o banco de dados do IdentIF com os campos: id, nome, siape, hash, tempo e os 128 pontos de identificação da face.

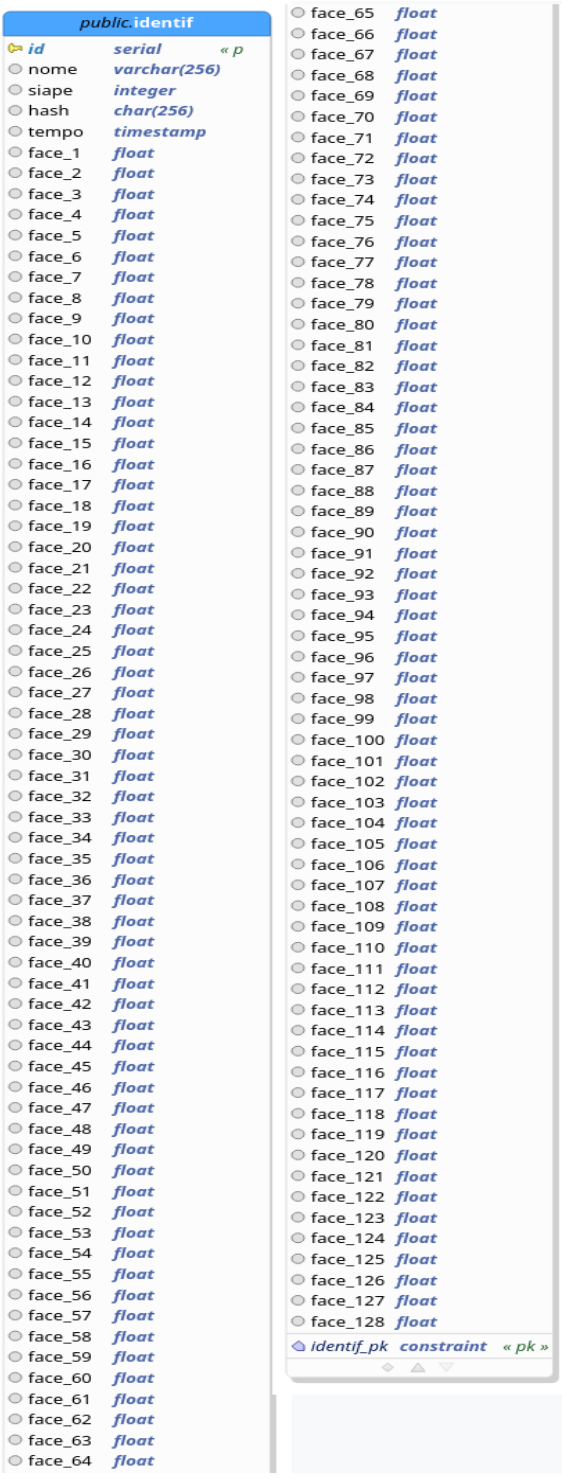

Figura 1. Banco de Dados - Autoria própria (2023).

O campo ID se refere a identificação do reconhecimento que aconteceu. Esse funciona como chave primária para buscas sendo um campo de valor único.

O campo nome se fere ao nome da pessoa que foi reconhecida pelo IdentIF.

O campo Siape se refere ao número de identificação do servidor público que foi identificado.

O campo Hash armazena o hash criptográfico, que é o resultado de uma função que dado uma entrada retorna um valor único em hexadecimal, esse é utilizado para o envio ao SUAP.

O campo tempo se refere a data e horário em que ocorreu a identificação. Essas informações são obtidas do sistema operacional Linux que estão sincronizadas por *NTP (Network Time Protocol)*. Esse protocolo é fundamental para garantir a exatidão necessária para um sistema de presença.

O reconhecimento facial acontece através do cálculo feito utilizando os seguintes campos:

face_1, face_2, face_3, face_4, face_5, face_6, face_7, face_8, face_9, face_10, face_11, face_12, face_13, face_14, face_15, face_16, face_17, face_18, face_19, face_20, face_21, face_22, face_23, face_24, face_25, face_26, face_27, face_28, face_29, face_30, face_31, face_32, face_33, face_34, face_35, face_36, face_37, face_38, face_39, face_40, face_41, face_42, face_43, face_44, face_45, face_46, face_47, face_48, face_49, face_50, face_51, face_52, face_53, face_54, face_55, face_56, face_57, face_58, face_59, face_60, face_61, face_62, face_63, face_64, face_65, face_66, face_67, face_68, face_69, face_70, face_71, face_72, face_73, face_74, face_75, face_76, face_77, face_78, face_79, face_80, face_81, face_82, face_83, face_84, face_85, face_86, face_87, face_88, face_89, face_90, face_91, face_92, face_93, face_94, face_95, face_96, face_97, face_98, face_99, face_100, face_101, face_102, face_103, face_104, face_105, face_106, face_107, face_108, face_109, face_110, face_111, face_112, face_113, face_114, face_115, face_116, face_117, face_118, face_119, face_120, face_121, face_122, face_123,

face_124, face_125, face_126, face_127, face_128.

Esses campos possuem seu valor atribuído em *float*(ponto flutuante) por quê os valores utilizados para o reconhecimento facial necessitam de uma boa precisão em seu cálculos por aproximação.

Esses 128 valores são comparados com os valores os quais a Inteligência Artificial do IdentIF através de treinamento de Redes Neurais aprendeu que a fazer a diferenciação de um servidor público em relação a outro.

Resultados e Discussão

A integração entre as tecnologias de hardware e software ocorreram bem devido ao uso de Linux, C++, OpenCV, QT, PostgreSQL e NVidia Jetson.

O SO(sistema operacional) Linux garantiu a estabilidade necessária para o desenvolvimento do sistema, garantindo que a depuração era necessária no software, pois nenhum era proveniente do SO.

A linguagem C++ possibilitou a performance desejável ao projeto. Ela também permitiu a interação entre OpenCV, QT e PostgreSQL.

A interface com os servidores públicos foi construída com a biblioteca QT, que facilita a criação de um ambiente visual agradável ao usuário final.

A OpenCV é a biblioteca de Visão Computacional que permitiu a criação da principal função do projeto, que é o reconhecimento facial.

O PostgreSQL foi o SGDB que permitiu a persistência dos dados.

A integração da comunicação entre a OpenCV, PostgreSQL e os métodos *HTTP(Hypertext Transfer Protocol)* necessários para o REST foram através do QT na linguagem C++.

Definir os campos do banco de dados foi um pouco mais complexo devido a compatibilidade com os dados advindos do reconhecimento facial, devido ao seu tipo.

Não houve dificuldade na interface REST.

Considerações Finais

O projeto se encaminhou bem, entretanto não houve retorno sobre como enviar os valores REST ao seu objetivo final.

A utilização de softwares livres e de código fonte aberto foram essenciais para o desenvolvimento desse projeto.

O hardware da NVidia Jetson permitiu que o processamento fosse realizado em GPU que facilita a paralelização e consequentemente diminui o tempo necessário para a identificação facial.

A integração entre essas tecnologias se mostrou promissora.

Agradecimentos

Agradeço ao meu pai Adevaldo Ribeiro da Silva, a minha mãe Leila Barbosa da Silva. Não posso deixar de mencionar meus orientadores, em especial os professores Eduardo Hiroshi Nakamura e Edson da Silva Castro, que me guiaram nesta pesquisa e também ao IFMS, CNPq e Fundect que viabilizaram o recurso para que a pesquisa fosse realizada. Agradeço pela oportunidade que me deram e pela paciência demonstrada ao longo de todo o processo de desenvolvimento. Cada um de vocês desempenhou um papel fundamental e sou profundamente grato por isso.

Referências

KAEHLER, Adrian; BRADSKI, Gary. Learning OpenCV 3: Computer Vision in C++ with OpenCV

Library. Sebastapol: O'Reilly Media, 2017.

LECHETA, R. Ricardo. Web Services RESTful. São Paulo: Novatec Editora, 2015.

MILANI, André. PostgreSQL: guia do programador. São Paulo: Novatec Editora, 2008.

SWAN, Tom. GNU C++ for Linux. Indinapolis: Que Corporation, 2000.

Database of IdentIF - Facial Identification Point

Abstract: *This project refers to the IndentIF system database, which uses Computer Vision for Facial Recognition of employees at the Federal Institute of Mato Grosso do Sul and recording their presence. To distinguish each person, this bank stores information about face details, time, cryptographic hash, name and name. The Software developed in C++ with the QT library and will, through the Artificial Intelligence technique, generate information that will be stored in the SGDB (Database Management System) PostgreSQL to later be sent to the SUAP (Unified Public Administration System) through REST (Representational State Transfer). Both the IdentIF and SGDB software were implemented and run on NVIDIA Jetson hardware, which runs Linux and allows the use of processing acceleration through the GPU (Graphics Processing Unit). The QT library has functions that communicate with the SGDB and enable interaction with the data contained therein.*

Keywords: *Artificial intelligence, C++, PostgreSQL*