Relatório do Projeto

Acsa Laiane Arcanjo Augusto
Gabriel Martins Spínola

[1]Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)

arcanjodagmar@gmail.com , martinsspn@gmail.com

Resumo. *Este documento descreve a implementação do projeto final da disciplina Linguagem de Programação 2 que consiste em implementar um sistema que identifique se há pessoa(s) em uma imagem e utilize essa informação para tomada de decisão.*

1. Introdução

O projeto trata-se de uma aplicação desenvolvida em java com o objetivo geral de Implementar um sistema que identifique se há pessoa(s) em uma imagem e utilize essa informação para tomada de decisão. Esse sistema foi desenvolvido em duas partes: a primeira, capaz de receber uma imagem como entrada, tratá-la e transformá-la em um vetor de características. Como também, a implementação do algoritmo "KNN" implementado em diferentes métricas (euclidiana, manhattan, chebychev), ou seja basicamente analisar dentre as imagens contidas no dataset disponibilizado as "k" imagens mais próximas (semelhantes) à respectiva imagem (de entrada), retornando o rótulo mais frequente. A segunda, uma interface gráfica (em JavaFx) que recebe um arquivo datacet.csv, uma imagem de entrada que será tratada e transformada (primeira parte), e escolher o tipo de métrica e o número k.

2. Implementação

Para implementar este projeto foi utilizada uma diversidade de classes conforme mostra o diagrama abaixo (figura 1).

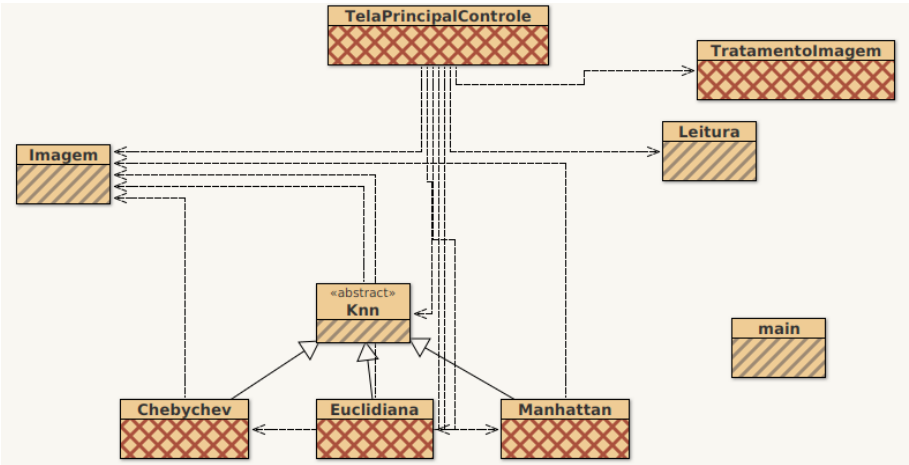

Figura 1. Diagrama da classe Imagem e Leitura

Essas foram divididas em 3 pacotes para facilitar o entendimento e promover a organização do código, da seguinte forma:

- Controle: Contém as classes de Imagem, Knn, Leitura, TelaPrincipalControle e TratamentoImagem, que serão discutidas posteriormente;
- Modelo: Contém as classes de Chebychev, Euclidiana e Manhattan, responsáveis por implementar a função knn para cada métrica;
- Visão: Contém a classe main que possui a função principal e o arquivo TelaPrincipal.fxml , ambos responsáveis por manipular a interface gráfica e passar certos dados dela para as outras classes.

2.1. Controle

Como dito anteriormente, este subgrupo consiste nas classes de Imagem, Knn, Leitura, TelaPrincipalControle e TratamentoImagem que se relacionam conforme os diagramas seguintes.

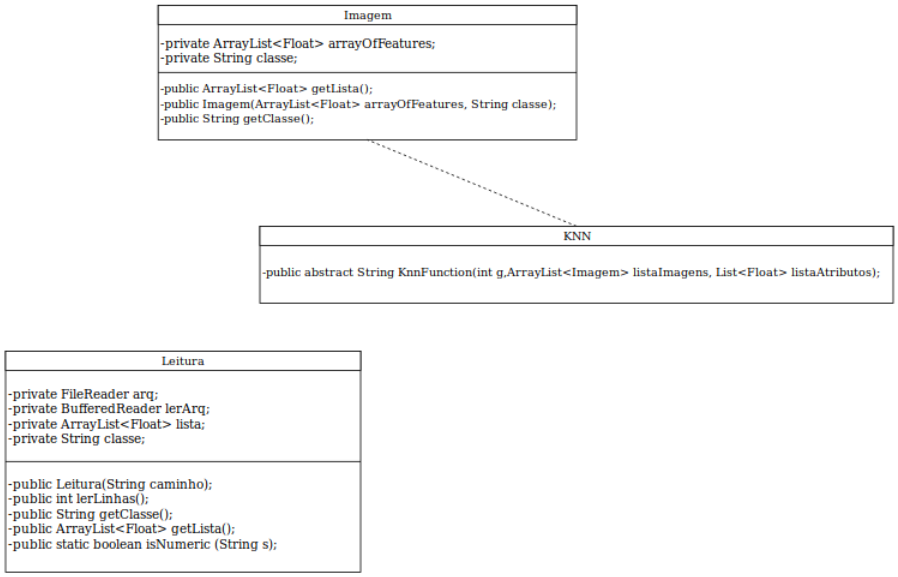

Figura 2. Diagrama das classes Imagem, Leitura e Knn

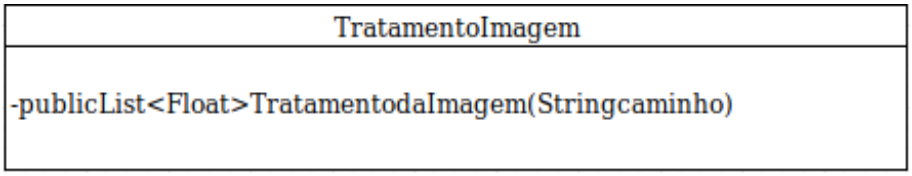

Figura 3. Classe de Tratamento das imagens

A classe **Imagem** é responsável por definir como uma imagem é representada, possuindo dois atributos: uma lista de números float, e uma string que informa se tem ou não pessoas.

Enquanto que a classe **Leitura** é responsável por ler os dados do dataset e transformar cada linha, gerando uma lista de números floats correspondentes aos atributos e guardando a o rótulo em uma string, para que posteriormente cada linha venha se tornar um objeto da classe Imagem, armazenados em uma lista.

Já a classe **TratamentoImagem** trata a imagem passada reduzindo o tamanho da sua imagem para 64 X 128 pixels, transformando cada imagem de entrada em um objeto

com as 1000 primeiras características disponíveis para o Histogram of Object Gradients (HOG) e realizando isso com a ajuda do OpenCV.

A classe abstrata **Knn** que possui a função que calcula as menores distâncias entre as imagens do dataset e a imagem passada atribuindo segundo esse resultado qual o rótulo(há pessoas, não há pessoas) da imagem passada, e que será implementada para cada métrica nas subclasses herdeiras.

E por fim, a classe **TelaPrincipalControle** é responsável por fornecer funcionalidade aos componentes criados no arquivo Interface.fxml bem como atuar como listenner dos botões presentes nessa interface.

2.2. Modelo

Este subgrupo é formado pelas classes herdeiras do Knn (Chebychev, Euclidiana e Manhattan).

O diagrama deste subgrupo pode ser visto abaixo: (Figura 4)

Como pode ser visto no diagrama, as classes desse grupo são herdeiras da classe abstrata Knn. Elas possuem funcionalidades similares, pois implementam a função do knn para cada métrica expecificada no nome da classe.

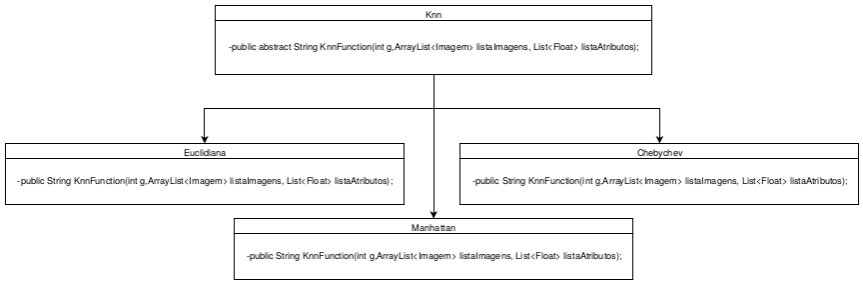

Figura 4. Diagrama da classe abstrata Knn e suas classes herdeiras

Contam com um método chamado **KnnFunction**, responsável por calular quais as **k** imagens do dataset possuem a menor distância com relação a imagem passada, verificando seu rótulo e atribuindo o rótulo mais frequente a imagem passada. No qual, a função para calcular as distâncias foi baseada nas fórmulas das métricas (Euclidiana (1), Manhattan (2), Chebychev (3)) abaixo.

$$\sqrt{\sum_{i=1}^{1000}(x_i - z_i)^2} \tag{1}$$

$$\sum_{i=1}^{1000}|x_i - z_i| \tag{2}$$

$$\mathbf{max}((z_1 - x_1), (z_2 - x_2), \cdots, (z_{1000} - x_{1000})) \tag{3}$$

Figura 5. Fórmulas das métricas

2.3. Visão

Este subgrupo é representado pelas classes main que possui a main e a classe TelaPrincipal, ambas responsáveis em fornecer uma interface gráfica interativa para o usuário assim como gerenciar requisições feitas por essa janela.

O diagrama de classes deste subgrupo é o seguinte:

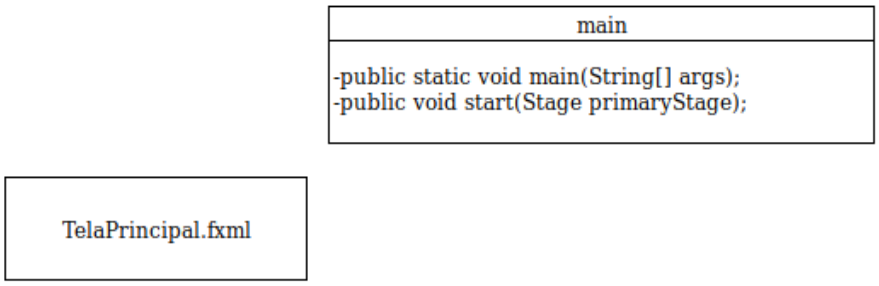

Figura 6. Diagrama das classes de main e TelaPrincipal

3. Fluxo de Execução

O fluxo de execução deste projeto é simples, ao executar a classe principal (**main**) o programa irá inicializar os componentes da interface e após inicializar todos, irá lançar a aplicação e passar o controle da aplicação para a classe **TelaPrincipalControle**.

Com o controle da aplicação em mãos, a classe **TelaPrincipalControle** irá aguardar o pressionamento de algum dos botões para realizar uma ação.

A interface gráfica conta com 3 áreas de texto visíveis, 4 botões, esses para pesquisar a imagem e o dataset, já tratando para só possibilitar a escolha de imagem **png** e datacet **csv**, outro botão para escolher a métrica, e o de verificar se há ou não pessoas. figuras (7,8,9).

4. Conclusão

Este trabalho apresentou a implementação de um sistema de identificação de pessoas, o qual, foi elaborado a fim de explorar e aprimorar o uso e conceitos de orientação à objeto, tais como modularização e padronização, herança e polimorfismo, classes abstratas, utilizando a linguagem java e os recursos de tratamento de exceções. Desenvolvendo um projeto baseado em conceitos de inteligência artificial e aprendizado de máquina, utilizando o algoritmo KNN.

Referências

https://opencv.org/

http://bit.ly/31X8RHp

https://openjfx.io/

https://stackoverflow.com/

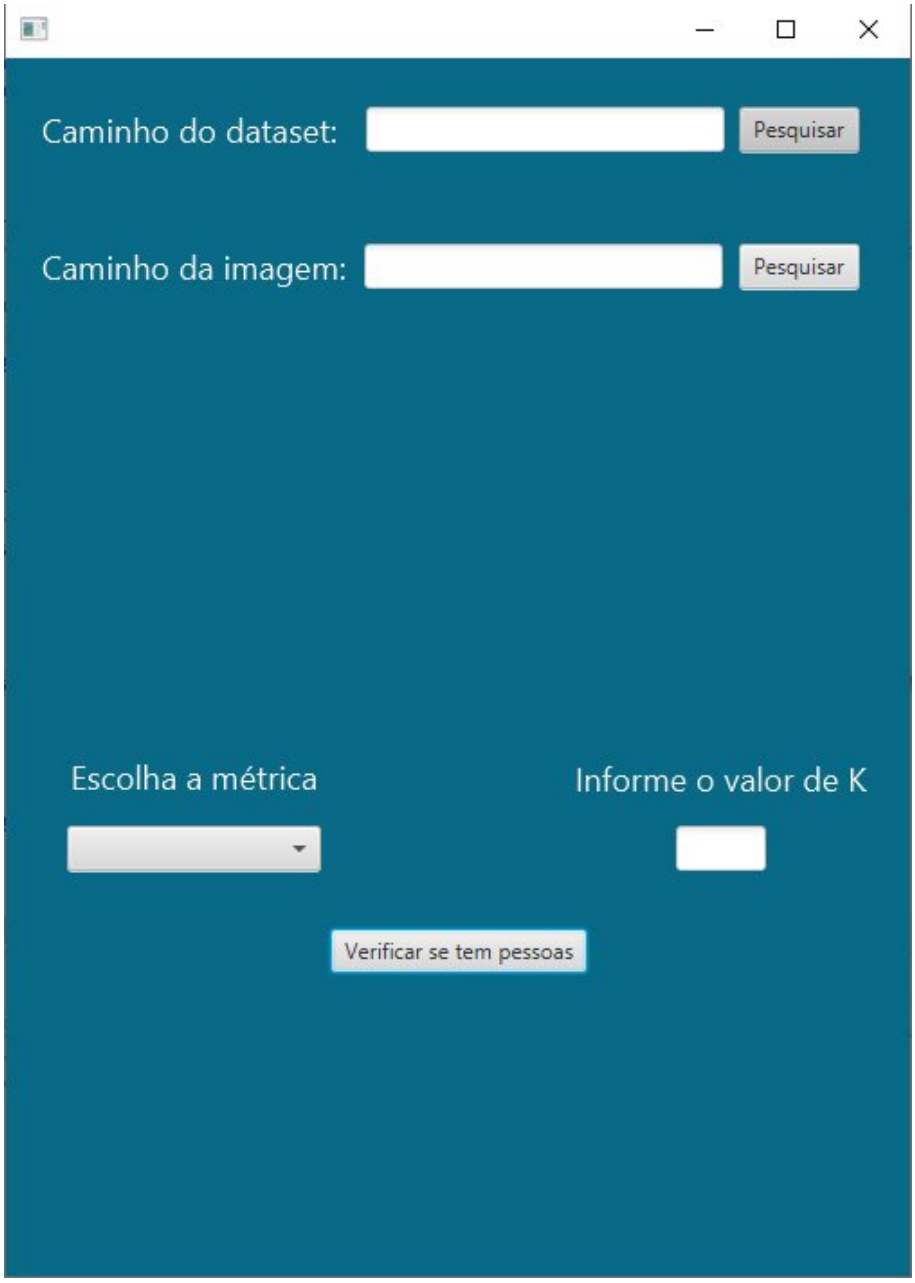

Figura 7. Tela inicial da interface

Figura 8. Interface após carregar imagem

Figura 9. Tela com a resposta