UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE

DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA – DEINFO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RANNIERY DIAS DE BRITO  
THIAGO ALVES BASTOS

**CLASSIFICADOR DE IMAGENS UTILIZANDO KNN PONDERADO**

Recife

2017

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO - UFRPE

**CLASSIFICADOR DE IMAGENS UTILIZANDO KNN PONDERADO**

Relatório da implementação de um sistema categorizador de imagens desenvolvido por Ranniery Dias de Brito e Thiago Alves Bastos como processo avaliativo da disciplina Inteligência Artificial, do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco (campus Recife), sob orientação do Prof. Dr. Valmir Macário Filho. O projeto pode ser baixado e analisado através do link: https://github.com/RannieryDias/Projeto\_IA.git

Recife

2017

**SUMÁRIO**

Introdução ----------------------------------------------------------------------------------- 4  
Objetivos ----------------------------------------------------------------------------------- 4  
Classes ----------------------------------------------------------------------------------- 5  
Vetor de Características ------------------------------------------------------------------- 5  
Algoritmo KNN ---------------------------------------------------------------------------- 5

Resultado --------------------------------------------------------------------------------- 6  
Conclusão --------------------------------------------------------------------------------- 7

# 

# INTRODUÇÃO

# A Inteligência Artificial (IA) é uma área da computação que tem como objetivo a “automação das atividades que nós associamos com o pensamento humano” (Bellman, 1978), tais como raciocínio, aprendizagem, percepção, entre outras. Portanto, é uma área que, apesar de nova, já está presente em diversas áreas da computação, o que faz com que seu aprendizado seja de grande importância.

# No presente projeto tínhamos por objetivo classificar imagens através de um algoritmo de IA chamado *K-Nearest Neighbors* (KNN). O mesmo é um algoritmo de aprendizagem baseado em instâncias (problemas resolvidos no passado). Essas instâncias correspondem a pontos em um espaço n-dimencional e as vizinhanças (*neighbors*) dessas instancias são definidas por uma função de distância (similaridade), ou seja, quanto menor a distância, maior a similaridade. Isto irá implicar na classificação de um novo exemplo, que será definida a partir dos vizinhos mais próximos, ou seja, de maior similaridade.

Para fazer a classificação foi utilizado como vetor de características o histograma, que é um conhecimento acerca das cores da imagem a ser classificada. Como entrada para nosso algoritmo usamos 7944 imagens, da base de dados *The CIFAR-10 dataset*, todas em tons de cinza, sendo 5296 imagens de treinamento e 2648 imagens de testes, todas de resolução 32x32. Tais imagens serão processadas em tons de cinza e em seguida serão classificadas pelo KNN em 10 classes.

Por fim, mostraremos os resultados da nossa classificação utilizando o KNN com a Distancia Euclidiana Ponderada (DEP). Para isso fizemos um comparativo da quantidade de acertos do KNN utilizando a DEP, a distância euclidiana, a distância manhattan e a distância manhattan ponderada.

**OBJETIVOS**

1. **Geral:**
   1. Classificar imagens em 10 diferentes classes por meio do algoritmo de inteligência artificial KNN.
2. **Específicos:**
   1. Utilizar a base de dados The CIFAR-10 dataset;
   2. Processar imagens em tons de cinza.
   3. Implementar o algoritmo KNN.
   4. Analisar os resultados obtidos.

**CLASSES**

As classes do nosso problema são os grupos de imagens que o KNN terá que classificar uma determinada imagem. As classes que utilizamos no nosso projeto são:

1. Airplane (Avião);
2. Automobile (Automóvel);
3. Bird (Pássaro);
4. Cat (Gato);
5. Deer (Veado);
6. Dog (Cachorro);
7. Frog (Sapo);
8. Horse (Cavalo);
9. Ship (Navio);
10. Truck (Caminhão)

As classes são mutuamente exclusivas, ou seja, não haverá uma imagem para 2 ou mais classes diferentes. No caso de Automóvel e Caminhão será classificado da seguinte maneira: Automóvel serão carros, SUVs, entre outros e, no caso de Caminhão, serão apenas caminhões de grande porte.

**VETOR DE CARACTERÍSTICAS**

O vetor de características do nosso projeto será o histograma da imagem em escala de cinza, ou seja, teremos um vetor de 256 posições onde as primeiras posições armazenam o número de pixels mais escuros (mais próximos de preto) até os mais claros (mais próximos do branco) e também a classe da imagem.

**ALGORITMO KNN**

Como já foi introduzido, o KNN é um algoritmo de inteligência artificial que visa fazer classificações baseado na distância entre o novo indivíduo e seus vizinhos em um espaço n-dimensional que, por estarmos trabalhando com o histograma de imagens em tons de cinza, o espaço tem 256 dimensões, que é o tamanho do nosso vetor de características.

Antes de realizar a classificação, é feita a leitura do arquivo “histogramas e classes”, para a criação dos vetores de treinamento e teste. O preenchimento desses vetores foi feito de duas maneiras, aleatória e ordenada, porém sempre mantendo a proporção de 2/3 das imagens, de cada classe, para o vetor de treinamento e de 1/3 das imagens, de cada classe, para o vetor de teste.

Para a realização da classificação, primeiro é calculada a distância entre a nova imagem e as demais já existentes (treinamento) e, a partir de um K pré-definido, classifica a nova imagem com a classe que se repete mais vezes entre as classes das K imagens de treinamento mais próximas da nova imagem.

A distância que estamos usando no nosso algoritmo é a Distancia Euclidiana Ponderada (DEP), que é calculada da seguinte maneira:

O cálculo do peso foi feito da seguinte maneira:

Ou seja, 1 dividido pela distância euclidiana.

**RESULTADO**

O algoritmo KNN conseguiu fazer a classificação, como será mostrado nas tabelas a seguir. Porém o mesmo não foi tão eficiente quanto esperávamos que iria ser. Entendemos que isso ocorreu por causa do vetor de características.

O algoritmo mostrou-se com comportamentos bem diferentes quando se definiam distâncias diferentes e com métodos de leitura diferentes, notamos que a distância DEP na leitura de arquivo aleatória, fez com que o KNN tivesse um melhor desempenho do que com as demais distancias, como mostrado nas tabelas a seguir.

K=5

|  |  |  |  |
| --- | --- | --- | --- |
| Distância Euclidiana | Distância Manhattan | Distância Manhattan Ponderada | Distância Euclidiana Ponderada |
| 100% em 4288 ms | 11% em 5047 ms | 11% em 9117 ms | 0% em 12771 ms |
| 100% em 2648 ms | 11% em 5021 ms | 11% em 9203 ms | 0% em 11960 ms |
| 100% em 4312 ms | 11% em 5028 ms | 11% em 9213 ms | 0% em 11532 ms |

Tabela 1.

Mostra os resultados obtidos em 3 execuções utilizando K=5 e leitura de arquivo aleatória.

K=5

|  |  |  |  |
| --- | --- | --- | --- |
| Distância Euclidiana | Distância Manhattan | Distância Manhattan Ponderada | Distância Euclidiana Ponderada |
| 20% em 6259 ms | 11% em 6847 ms | 11% em 10678 ms | 20% em 12571 ms |
| 20% em 6286 ms | 11% em 6765 ms | 11% em 12210 ms | 20% em 12585 ms |
| 20% em 6195 ms | 11% em 7220 ms | 11% em 11597 ms | 20% em 12276 ms |

Tabela 2.

Mostra os resultados obtidos em 3 execuções utilizando K=5 e leitura de arquivo ordenada.

K=57

|  |  |  |  |
| --- | --- | --- | --- |
| Distância Euclidiana | Distância Manhattan | Distância Manhattan Ponderada | Distância Euclidiana Ponderada |
| 100% em 4328 ms | 11% em 6990 ms | 11% em 7005 ms | 0% em 10200 ms |
| 100% em 4284 ms | 11% em 7125 ms | 11% em 6717 ms | 0% em 12036 ms |
| 100% em 4310 ms | 11% em 6872 ms | 11% em 6820 ms | 0% em 11219 ms |

Tabela 3.

Mostra os resultados obtidos em 3 execuções utilizando K=57 e leitura de arquivo aleatória.

K=57

|  |  |  |  |
| --- | --- | --- | --- |
| Distância Euclidiana | Distância Manhattan | Distância Manhattan Ponderada | Distância Euclidiana Ponderada |
| 20% em 6248 ms | 11% em 6986 ms | 11% em 11410 ms | 20% em 12351 ms |
| 20% em 6201 ms | 11% em 7080 ms | 11% em 11365 ms | 20% em 12301 ms |
| 20% em 6111 ms | 11% em 7205 ms | 11% em 11373 ms | 20% em 12534 ms |

Tabela 4.

Mostra os resultados obtidos em 3 execuções utilizando K=57 e leitura de arquivo ordenada.

K=127

|  |  |  |  |
| --- | --- | --- | --- |
| Distância Euclidiana | Distância Manhattan | Distância Manhattan Ponderada | Distância Euclidiana Ponderada |
| 100% em 4287 ms | 11% em 7066 ms | 11% em 11864 ms | 0% em 11398 ms |
| 100% em 4325 ms | 11% em 7114 ms | 11% em 12286 ms | 0% em 11108 ms |
| 100% em 4292 ms | 11% em 7052 ms | 11% em 11505 ms | 0% em 12554 ms |

Tabela 4.

Mostra os resultados obtidos em 3 execuções utilizando K=127 e leitura de arquivo aleatória.

K=127

|  |  |  |  |
| --- | --- | --- | --- |
| Distância Euclidiana | Distância Manhattan | Distância Manhattan Ponderada | Distância Euclidiana Ponderada |
| 20% em 6092 ms | 11% em 7084 ms | 11% em 11504 ms | 20% em 12418 ms |
| 20% em 6247 ms | 11% em 7097 ms | 11% em 11357 ms | 20% em 12346 ms |
| 20% em 6172 ms | 11% em 7187 ms | 11% em 11304 ms | 20% em 12266 ms |

Tabela 4.

Mostra os resultados obtidos em 3 execuções utilizando K=127 e leitura de arquivo ordenada.

**CONCLUSÃO**

Ao final do projeto, conseguimos desenvolver o algoritmo proposto (KNN), trabalhamos com as imagens em tons de cinza e utilizamos como vetor de característica o histograma.

Porém, obtivemos nos nossos testes resultados abaixo do esperado, com apenas 20%, no máximo, de acertos, como mostrado nas tabelas da sessão anterior. Isso ocorreu pela escolha do histograma como vetor de característica de imagens em tons de cinza.

O histograma não é um vetor de características adequado para a classificação de imagens em tons de cinza com o KNN, pois é pouca informação para fazer muitas diferenciações, já que estamos usando 10 classes.

Concluímos que mesmo conseguindo resultados melhores com a distância escolhida (DEP), talvez esses resultados tivessem sido melhores se tivéssemos utilizado o histograma de imagens em RGB, como o vetor de características, ou com a diminuição de classe, ou seja, ao invés de diferenciar 10 classes, diferenciaria apenas 2 classes.