

Relatório do projeto da disciplina de Introdução à Teoria da Informação Implementação de um reconhecedor de padrões baseado no compressor Lempel-Ziv-Welch(LZW) utilizando a linguagem de programação C++

Nathan Rodrigues Tavares de Lima

Universidade Federal da Paraíba - Centro de Informática

Depto. de Sistemas para Computação

Engenharia da Computação

João Pessoa, Paraíba

Email: nathanlima@eng.ci.ufpb.br

Wesley Dezidério da Silva

Universidade Federal da Paraíba - Centro de Informática

Depto. de Sistemas para Computação

Engenharia da Computação

João Pessoa, Paraíba

Email: wesleydeziderio@eng.ci.ufpb.br

Abstract—This report seeks to present and discuss the results obtained during the development of the second project from the Introduction to Information Theory subject made at Universidade Federal da Paraíba. Using the C++ programming language, we implemented a pattern recognizer based on Lempel-Ziv-Welch(LZW) compressor/decompressor.

November 17, 2022

Resumo—Este relatório busca apresentar e discutir os resultados obtidos durante o desenvolvimento do segundo projeto da disciplina de Introdução à Teoria da Informação realizado na Universidade Federal da Paraíba. Utilizando a linguagem de programação C++, realizamos a implementação de um reconhecedor de padrões baseado no compressor/descompressor Lempel-Ziv-Welch (LZW).

17 de novembro, 2022

1. Introdução

Na inteligência artificial um, **reconhecedor de padrões** é o processo de reconhecer padrões através da classificação de um conhecimento já obtido, em que seu potencial pode ser aplicado em várias áreas e problemas cotidianos, tais como reconhecimento de vozes, reconhecimento de pessoas, entre outros.

Utilizando um banco de dados de faces de pessoas em preto e cinza (ORL Database of faces) no formato de arquivo sem compressão .pgm (Portable Gray Map) e utilizando a linguagem de programação C++ desenvolvemos esse reconhecedor, não utilizando Inteligência Artificial, mas sim, como foi dito previamente utilizando um compressor e descompressor multimídia.

Podemos definir um arquivo .pgm como uma imagem em tons de cinza $I(x, y)$ (ex: imagem de ultrassom, fatia tomográfica) e bidimensional é uma matriz com M linhas e N colunas. Nesta matriz cada elemento $I(x, y)$, $x = 0, 1, \dots, N-1$ e $y = 0, 1, \dots, M-1$, é chamado pixel (uma abreviação de picture elements), e o valor de $I(x, y)$ é proporcional ao brilho da imagem neste ponto, indo de 0 (preto) até o valor máximo 255 (branco), quantizados em uma escala de níveis de cinza.

Além disso, esse é seu formato de arquivo:

P2 (Cabeçalho)

Largura e Altura

Valor máximo de intensidade

Valor do Pixel

Dessa forma, descartamos o cabeçalho e trabalhamos apenas com os valores de pixel da imagem. Isso foi realizado visando diminuir as chances de problemas durante a compressão/descompressão das imagens.

O banco de treinamento e teste foram gerados de forma aleatória, de forma que foram selecionadas na proporção de 1:1, 1 imagem para treino e 1 imagem para a classificação. Para cada rosto separados por pastas, foram selecionadas 5 imagens para treino e 5 imagens para testes.

2. Estaticidade do dicionário

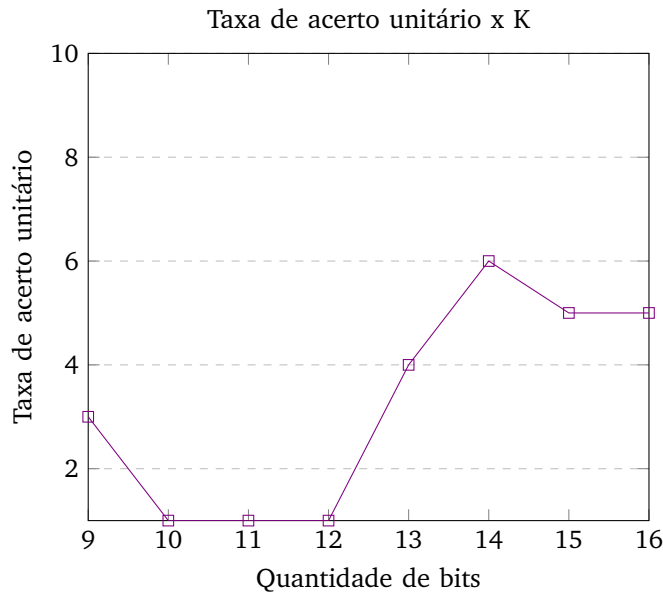
Depois que o dicionário atinge o seu tamanho máximo, ele **permanece estático**. Salientando que seu valor máximo depende do 2^k , temos como valor máximo $2^{16} = 65536$

3. Taxa de Acerto x K

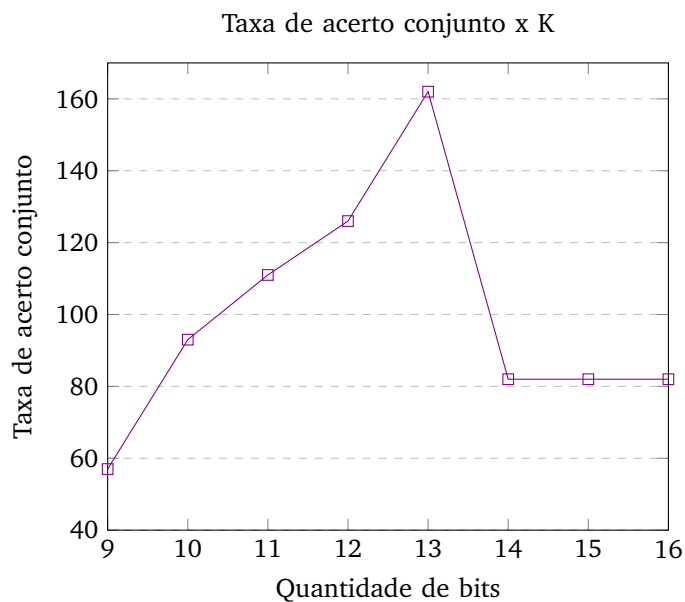
Foi possível testar para todos os valores de K (quantidade de bits), onde para estimar a distância usamos a quantidade de índices do arquivo comprimido.

A taxa de acerto foi dividida em duas seções. O acerto unitário, que é quando a predição retorna uma única previsão correta, e nos casos onde haver mais de uma predição ou apenas uma incorreta, será contabilizada como erro. E o acerto conjunto, que é quando os acertos unitários são contabilizados e uma predição que contém mais de uma previsão e dentre elas a correta, será contabilizado como acerto.

3.1. Taxa de acerto unitário

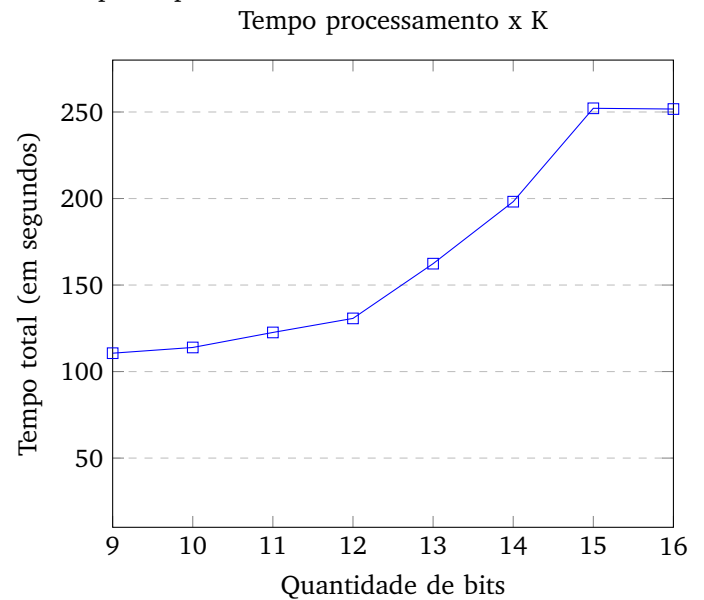


3.2. Taxa de acerto conjunto



4. Curva de Tempo x K

Para a curva de tempo x K, realizamos o somatório de todo o tempo de processamento.



5. Resultados

Acreditamos que a taxa de acerto foi um pouco prejudicada pela forma que abrimos o arquivo, como um array de bits. Sendo assim, houveram alguns caracteres que não foram bem representados, diminuindo a acurácia da nossa previsão.

No entanto, apesar do problema citado, o trabalho em questão permitiu que pudéssemos expandir os conhecimentos apresentados em sala de aula.

Referências

- [1] IME-USP. MAC110 Introdução à Computação, 2014. URL: http://www.vision.ime.usp.br/~pmiranda/mac110_1s14/EPs/ep02/pgm_ppm.pdf.