UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE INFORMÁTICA

ENGENHARIA DE COMPUTAÇÃO

Trabalho Final - Introdução à Teoria da Informação
PPM - Reconhecimento de Padrões

Arthur Curty Vieira - 11506859
George Nunes de Moura Filho - 11328786
José Eugênio Carvalho De Souza - 11506862
Lucas Rincon - 11400992
Thiago Gonzaga Gomes - 11504760

Orientador: Prof. Dr. Derzu Omaia

João Pessoa – 07 de maio de 2019

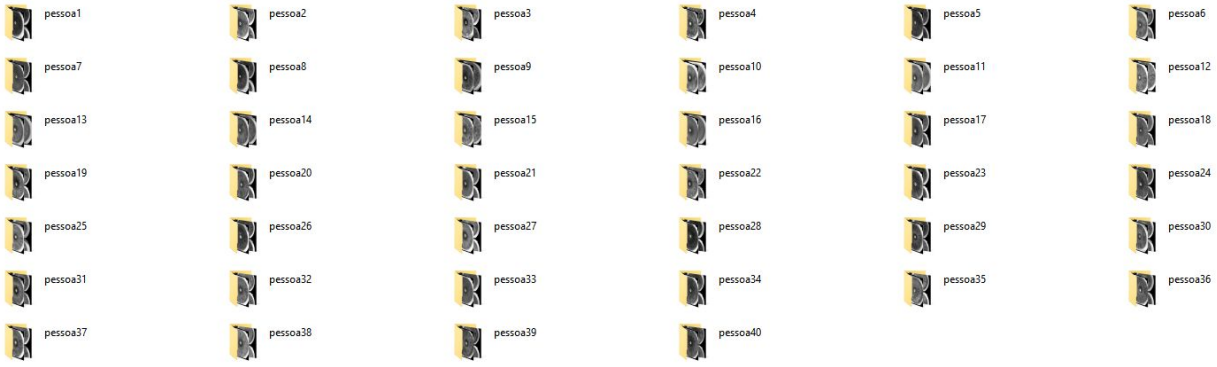
Sumário

Introdução

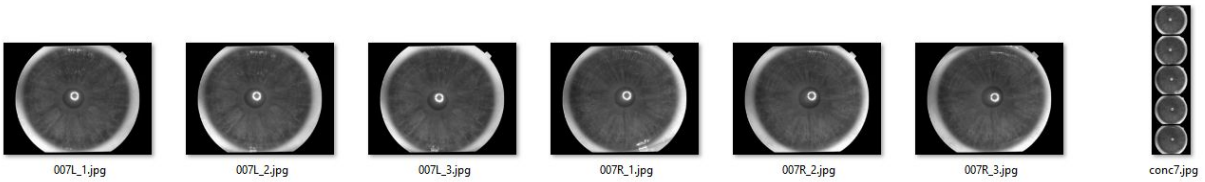
Este relatório abordará o desenvolvimento da implementação de um reconhecedor de padrões baseado no PPM.

Para a implementação, foi utilizado um banco de dados pré-disponibilizado pelo professor, relacionado as íris (*Iris Database Palacký University*), que consiste de um grupo de 40 *subjects* com 6 fotos cada (3 fotos da íris do olho esquerdo e 3 fotos da íris do olho direito).

Um pré-tratamento foi realizado no banco, onde foi feita uma divisão das amostras em grupos, um para cada *subject* do banco.



Outro tratamento realizado, foi o da aplicação de escala de cinza em todas as imagens



E por fim, para a facilitação da criação e percorrimento da árvore do PPM, foram concatenadas as 5 imagens de treinamento, empilhadas na vertical, deixando sempre de fora a primeira imagem de cada grupo, que servirá como modelo de teste.

Desenvolvimento

A primeira etapa da implementação foi a criação das árvores de treinamento utilizando o PPM nas imagens concatenadas. Foram criadas 40 árvores para os 9 níveis de contexto, totalizando em 360 árvores. Para cada compressão das imagens concatenadas, foi gerada sua árvore de treinamento correspondente, juntamente com algumas informações disponibilizadas durante a compressão.

Como por exemplo, na amostra da pessoa #4, pode-se ver as informações de *output* da compressão gerada pelo ppm, tais quais, o tamanho do arquivo de entrada (em bytes), o tempo total de compressão (em segundos), o tamanho de saída da compressão, o tamanho médio (relação bits/símbolo), a entropia (relação bits/símbolo) e a razão de compressão.

Pessoa 4	Tamanho Entrada	Tempo Total	Tamanho Saída	Tamanho Médio(bits/símbolo)	Entropia (bits/simbolo)	RC
k = 0	2496	6.357	2488	7.974	7.974	1.003 : 1
k = 1		10.864	2481	7.952	7.951	1.006 : 1
k = 2		12.228	2659	8.521	8.520	0.939 : 1
k = 3		13.554	2664	8.538	8.537	0.937 : 1
k = 4		13.749	2664	8.538	8.538	0.937 : 1
k = 5		14.375	2665	8.539	8.538	0.937 : 1
k = 6		14.101	2665	8.539	8.539	0.937 : 1
k = 7		14.607	2665	8.540	8.540	0.937 : 1
k = 8		31.748	2665	8.540	8.540	0.937 : 1

Após a criação das 360 árvores de treinamento, foram criadas as 360 árvores de classificação, a partir do arquivo que ficou de fora, no caso, o primeiro arquivo de cada grupo (*00*L_1.jpg*).

Com elas, temos as mesmas saídas obtidas na criação das árvores de treinamento:

Pessoa 4	Tamanho Entrada	Tempo Total	Tamanho Saída	Tamanho Médio(bits/símbolo)	Entropia (bits/símbolo)	RC
k = 0	503	1.118	502	7.990	7.989	1.001 : 1
k = 1		1.930	504	8.019	8.018	0.998 : 1
k = 2		2.134	524	8.340	8.338	0.959 : 1
k = 3		2.231	525	8.343	8.341	0.959 : 1
k = 4		2.345	525	8.343	8.342	0.959 : 1
k = 5		2.420	525	8.344	8.343	0.959 : 1
k = 6		2.434	525	8.345	8.343	0.959 : 1
k = 7		2.538	525	8.345	8.344	0.959 : 1
k = 8		2.630	525	8.346	8.345	0.959 : 1

Após a obtenção das árvores de treinamento e de classificação, um modelo de IA foi aplicado para a realização da métrica de classificação. Um tamanho médio das árvores de treinamento foi obtido dividindo por ***5*** (cinco imagens concatenadas formando uma árvore) o tamanho total das mesmas e comparado n-n a cada nível de contexto.

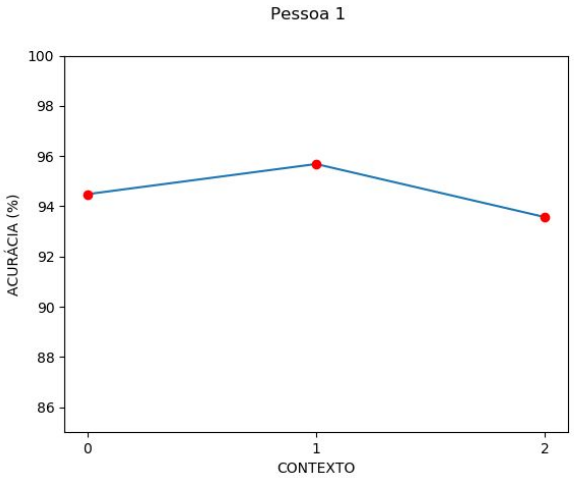
Após a obtenção desse tamanho médio, o valor obtido foi subtraído pelo tamanho total da árvore de classificação correspondente ao nível comparado, em questão, da árvore de treinamento. E uma função de *abs()* foi aplicada no valor resultante.

Com isso, foi possível comparar a acurácia da compressão nos dando a métrica: quanto menor o tamanho da árvore de classificação correspondente e comparada com a árvore de treinamento no mesmo nível de contexto, melhor sua compressão.

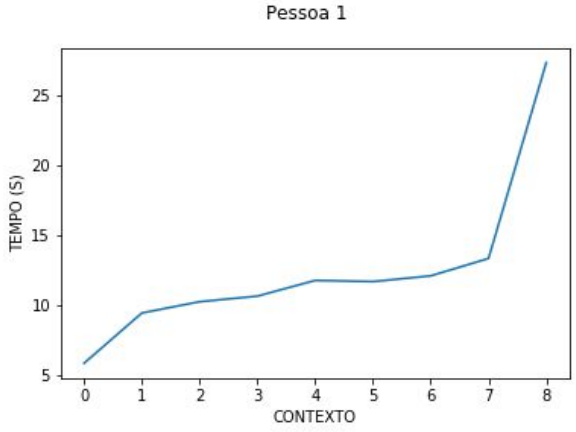
Para achar a acurácia da classificação, cada imagem foi convertida em uma linha de um *dataframe* e o ***random forest*** foi utilizado como classificador. Os eixos *x* e *y* foram fixados de forma que utilizassem a menor porcentagem gerada pelo classificador, que foi 85%.

Resultados

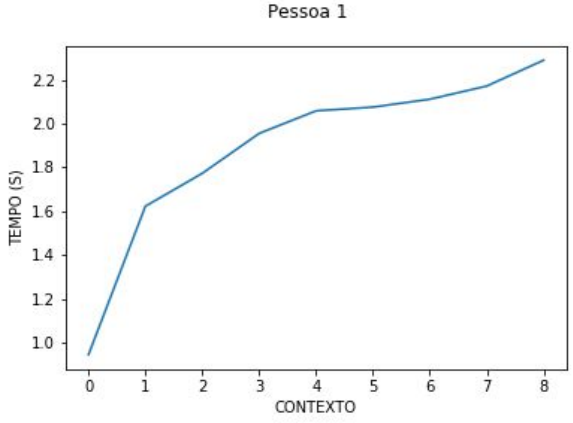
Como resultados, foram plotados gráficos relacionados à ***taxa de acurácia*** e ao ***tempo de processamento*** para os 9 níveis de contexto.



Acuráxia x Contexto



Tempo x Contexto (Árvore de Treinamento)



Tempo x Contexto (Árvore de Classificação)

Conclusão

Com a aplicação dos modelos e métricas juntamente com o PPM, é possível concluir que o PPM é sem dúvida um excelente método de compressão, sem perdas, que consegue atingir ótimas ratios de compressão e com um número médio de bits por caracteres muito próximos da entropia, uma vez que para além de tirar partido de um excelente modelo estatístico, tendo em conta sempre o contexto em que o caractere aparece no texto, faz uso do codificador aritmético adaptativo. Como foi demonstrado com as variantes do PPM, quanto mais complexo se torna o algoritmo, de modo a obter melhores ratios de compressão, mais lenta a codificação se torna.

Com a aplicação dos níveis de contexto e separação por “frequência” de repetição de bits/caracteres (entropia), o PPM é de grande serventia e ajuda para o reconhecimento de padrões dentro de um dado arquivo a ser comprimido.