

Reconhecimento de Padrões

Reconhecimento de Script Usando Textura

Mateus Torres¹, Emanuel Mazzer¹, João Gris¹, Leonardo Corra¹

¹Departamento de Ciências da Computação
Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão – PR – Brasil

{torres, joaogris, leonardoribeiro, mazzer}@alunos.utfpr.edu.br

Abstract. *This paper describes the development of a pattern recognition algorithm following the classical structure, that is, acquisition, preprocessing, segmentation, feature extraction and classification. The purpose of this system is to identify, given an image referring to the digitization of a handwritten letter, in which language it was written. For the extraction of the characteristics, we use the descriptors, LBP images (Local Binary Patterns) and LPQ (Local Phase Quantization). The classification algorithms chosen for this problem are the Random Forest and the Nearest Centroid. In general, the developed system presented accuracy rates of approximately 93% in the identification of the class of samples.*

Resumo. *Este artigo descreve o desenvolvimento de um algoritmo de reconhecimento de padrões seguindo a estrutura clássica, isto é aquisição, pré-processamento, segmentação, extração de características e classificação. O objetivo desse sistema é identificar, dada uma imagem referente a digitalização de uma carta manuscrita, em qual idioma esta foi escrita. Para extração das características foram utilizados os descritores de imagens LBP (Local Binary Pattern) e LPQ (Local Phase Quantization). Os algoritmos de classificação escolhidos para este problema o Random Forest e o Nearest Centroid. Em geral o sistema desenvolvido apresentou taxas de acerto de aproximadamente 93% na identificação da classe das amostras.*

1. Introdução

Reconhecimento de padrões é uma modalidade que vem sendo aplicada como base para classificação de elementos nos mais diversos contextos. O modelo mais popular para desenvolvimento de sistemas de reconhecimento de padrões é definidos pelas etapas de aquisição, pré-processamento, segmentação, extração de características e finalmente de classificação [Martins , Bertolini et al. 2013].

Neste artigo será apresentado o desenvolvimento de um algoritmo que tem como finalidade, dada uma entrada de dados referente a digitalização de cartas escritas manualmente, a identificação de qual idioma a entrada sendo avaliada pertence, sendo as possíveis classificações de idioma Bangla, Oriya, Persian e Roman.

Para este trabalho não foi necessário realizar a etapa de aquisição pois as amostras utilizadas nos conjuntos de treino e teste já haviam sido publicadas em outros trabalhos. A extração de características das amostras foi executada utilizando os descritores de imagens LBP (*Local Binary Pattern*) e LPQ (*Local Phase Quantization*). Os algoritmos de classificação escolhidos para este problema foram o *Random Forest* e o *Nearest Centroid*.

2. Amostras

A base de amostras utilizadas neste trabalho é constituída por um conjunto de imagens composto por um total de 222 elementos. Esses elementos correspondem a digitalização de cartas que foram escritas manualmente e em 4 idiomas diferentes, sendo eles: Bangla, Oriya, Persian e Roman, formando assim grupos de cartas com 66, 50, 16, 90 arquivos de amostra respectivamente. As amostras utilizadas neste trabalho também passaram por um pré-processamento proferido por outros autores onde foram removidos os espaços em branco entre as linhas e as palavras de forma a concatenar as escritas detectadas na imagem.

3. Pré-processamento

Quando se fala em reconhecimento de padrões normalmente uma das etapas mais importantes é o pré-processamento, visto que muitas vezes isolar, realçar ou agrupar características logo nessa etapa pode fazer com que o classificador seja mais eficiente em distinguir as características da classe, tornando o processo de reconhecimento mais eficiente. Porém também é possível que um pré-processamento que altere muito a amostra, também prejudique o desempenho do classificador.

Neste trabalho o conjunto de amostras já havia sido pré-processada anteriormente, de forma que as imagens apresentassem apenas a representação da textura das cartas originais. Esse pré-processamento também tratou a ocorrência de ruídos nas imagens tornando desnecessário um processo de remoção do mesmo. Dessa forma, para a execução deste trabalho foi necessário apenas remover imagens completamente em branco, isto é, sem nenhuma escrita detectada, e também a remoção de bordas brancas verificadas nas amostras.

3.1. Remoção das imagens sem conteúdo

Fez-se necessário a remoção de imagens onde não fôra detectado a ocorrência de nenhuma escrita, pois se estas imagens fossem processadas pelos métodos que serão descritos nos tópicos seguintes provavelmente ocorreriam erros durante a classificação das classes, já que se essas imagens fossem utilizadas como conjunto tanto de treino quanto de teste, o classificador interpretaria o vetor de características da imagem, neste um vetor vazio, como sendo uma característica representante da classe sendo processada.

3.2. Remoção da parte branca das imagens

O processo de remoção das bordas brancas verificadas nas imagens do conjunto de amostra foi aplicado a fim de diminuir a quantidade de memória necessária para armazenar os arquivos referentes a estas amostras. Além disso, a remoção desses dados também influencia positivamente no tempo de processamento de cada imagem, visto que sem as bordas brancas, existem menos *pixels* a serem processados pelo algoritmo que vai interpretar a imagem.

4. Segmentação

O processo de segmentação subdivide uma amostra em partes ou objetos constituintes. O nível até o qual essa subdivisão deve ser realizada depende do problema em questão. A segmentação é uma das tarefas mais difíceis do ponto de vista computacional. Se ela falha, a maioria dos processos subsequentes é comprometido. Para este trabalho a segmentação foi realizada de duas formas diferentes. A primeira delas foi gerando um número de blocos fixo, isto é, uma grade, que sobrepôs e dividiu cada uma das amostras da base utilizada neste projeto. A segunda abordagem foi gerar blocos de tamanhos iguais ao da abordagem anterior também aplicadas sobre a imagem em posições aleatórias, de forma a dividi-la.

4.1. Blocos Fixos

A segmentação com blocos fixos foi aplicada de forma que cada imagem fosse dividida em nove blocos de mesma largura e, de altura igual a altura da imagem original. Anteriormente também foram feitos testes com utilizando a divisão com 50 blocos por imagem, porém, a segmentação não se mostrou viável, pois gerou uma excessiva quantidade de blocos completamente em branco.

4.2. Blocos Aleatórios

Na segmentação com blocos aleatórios o tamanho dos blocos foi definido da mesma forma que foram calculados o tamanho dos blocos fixos. Isto é, com a mesma altura da imagem e largura igual a largura da imagem dividida por nove. O ponto de início foi sorteado aleatoriamente, com o intuito de gerar blocos em posições aleatórias da imagem.

5. Extração das características

Em reconhecimento de padrões a extração de características é uma forma de redução dos dados a serem processados. Quando a entrada de dados para um determinado algoritmo é muito grande e notavelmente redundantes é preciso transformá-la em um conjunto reduzido de características melhor representativo para que seja viável o processamento dos dados.

Essa transformação é denominada extração de características. Caso as características extraídas sejam adequadamente escolhidas espera-se que o conjunto de características represente as partes mais relevantes das informações da amostra para se executar a tarefa desejada ao invés de se usar os dados de entrada na íntegra. A extração de características é uma fase fundamental para o sucesso na etapa da classificação das imagens.

Em resumo a extração de características envolve simplificar o conjunto de dados necessários para descrever um grande conjunto com mais precisão. Deste modo, neste trabalho foram extraídas, para cada uma das amostras, um conjunto de 256 características. Essa extração foi executada por meio de dois extratores diferentes onde cada um deles também foi executado sobre os dois tipos de segmentação descritos anteriormente. A seguir serão apresentados os descritores utilizados nesse trabalho com o propósito de classificar imagens de cartas escritas manualmente de acordo com seus idiomas. Nesta etapa foram testados dois descritor a fim de selecionar qual se adéqua melhor ao problema de reconhecimento de padrão estudado neste trabalho, sendo eles o LBP e o LPQ.

5.1. Descritor LBP

O LBP (Local Binary Pattern) é um descritor de textura utilizado para análise e extração de característica de imagens. Quando aplicado a uma imagem, o LBP gera uma representação numérica das características gráficas desta imagem. O LBP descreve a associação de correspondência do nível de cinza de um pixel e seus vizinhos. O cálculo do LBP associado a um pixel é feito a partir de um conjunto de p amostras igualmente espaçadas e distribuídas sobre uma circunferência de raio r com centro sobre o pixel [Dartora and Oliveira]. Desta forma são gerados números binários de oito dígitos que são utilizados para computar o vetor de 256 características de cada amostra que posteriormente será utilizado no classificador.

5.2. Descritor LPQ

O LPQ (Local Phase Quantization) é um extrator bem conhecido baseado em componentes de histograma. Ele baseia-se na invariante de “borrão” do espectro de fase de Fourier. Ele utiliza a informação da fase local extraída de uma transformação de Fourier de curto prazo calculada sobre uma vizinhança retangular colocada sobre cada pixel da imagem. No LPQ, apenas quatro coeficientes complexos são considerados, sendo esses correspondentes às frequências 2D da imagem [Ahonen et al. 2008].

6. Classificação

Um classificador tem como função principal atribuir padrões a classes distintas. O desenvolvimento de classificadores envolve instanciar alguma forma geral do modelo, ou forma de classificação de modelo que se possa usar padrões de treinamento para aprender ou estimar os parâmetros desconhecidos de um modelo. Cada classificador generaliza de forma diferente um mesmo modelo, de acordo com se limiar de decisão, em geral, a escolha de um classificador base-a-se no conjunto de validação mais eficaz apresentado durante uma classificação.

A classificação dos dados geralmente é baseada na disponibilidade de um conjunto de padrões que foram anteriormente classificados, sendo este denominado como conjunto de treino. O resultado do treinamento é caracterizado como um aprendizado supervisionado. O aprendizado também pode ser não supervisionado, nesta abordagem o sistema não recebe informações a priori dos padrões, estabelecendo então as classes dos padrões através de análise de padrões estatísticos.

Neste trabalho utilizou-se o método de classificação supervisionado. O conjunto de imagens da base foi dividido em dois conjuntos diferentes, sendo 75% das imagens utilizadas como conjunto de treino e 25% para execução de teste. A execução do classificador *Random Forest*, referente ao treino, foi realizada utilizando uma floresta com 300 árvores. Nas subseções a seguir será descrito mais detalhadamente o funcionamento dos classificadores escolhidos para este trabalho.

6.1. Random Forest

O *Random Forest* é um algoritmo que gera um conjunto com vários classificadores. Basicamente ele é a combinação de árvores de predição, onde cada árvore depende de valores de um vetor aleatório como amostra independente e com a mesma distribuição de todas as demais árvores da floresta. Cada árvore avalia uma característica distinta. Ao fim

da execução o resultado obtido de cada árvore é combinado e a classificação da classe é determinada. Essas árvores de decisão funciona com base nos registros do conjunto de treinamento, gerando uma estrutura de árvore e, a partir desta árvore, classificando a amostra desconhecida sem necessariamente testar todos os valores dos seus atributos. A Figura 1 apresenta uma visualização de como funciona esse algoritmo. Neste exemplo temos n árvores instanciadas onde cada uma delas representa um conjunto de características e quando chegasse ao nó folha é determinado a qual classe esse conjunto corresponde. Como os resultados podem variar devido ao conjunto aleatório de características ao fim execução é determinado a classe a qual a amostra pertence por meio da realização de uma votação.

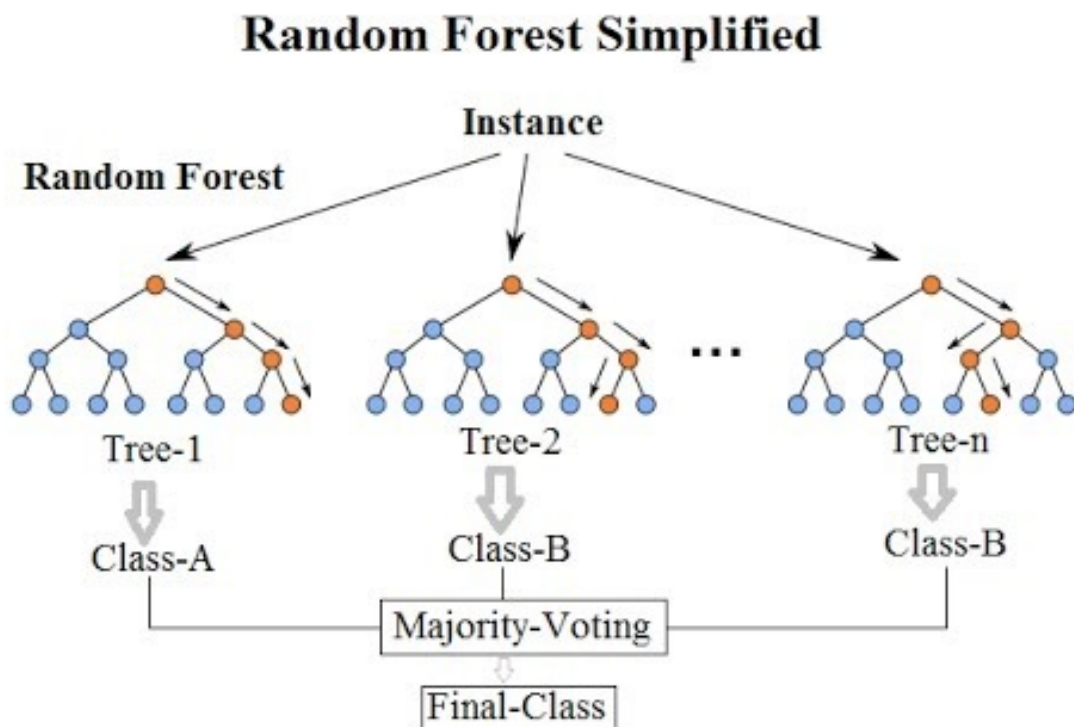


Figura 1: Classificador *Random Forest*

6.2. Nearest Centroid

Em aprendizagem de máquina, o classificador *Nearest Centroid* é um modelo de classificação que atribui as características o rótulo da classe associada de acordo com a execução de treinamento, a média ou centroide representa a classe mais próxima da característica analisada. Na Figura 2 é possível observar a aplicação do *Nearest Centroid*. Cada cor de círculo representa elementos de classes distintas. O classificador determina a classe da amostra observando os vizinhos correspondentes as características analisadas.

Incremental nearest centroid classifier

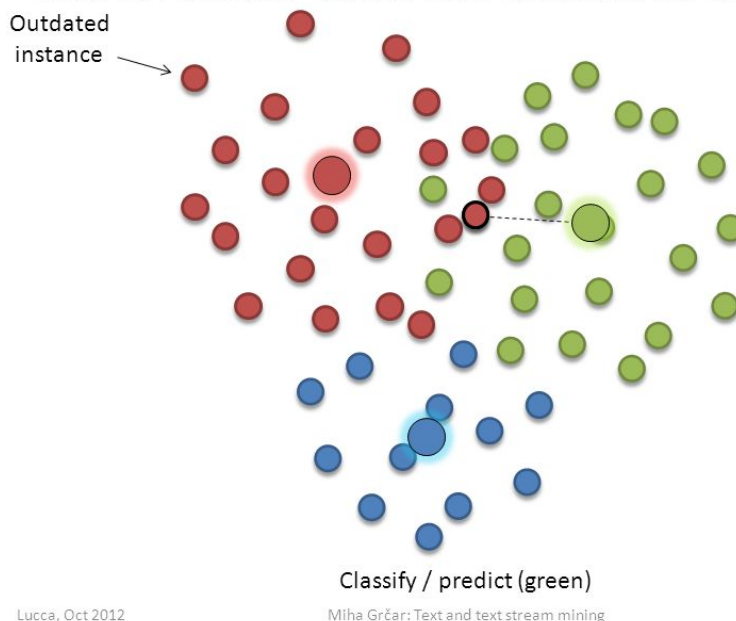


Figura 2: Classificador *Nearest Centroid*

7. Resultados

Nesta seção serão apresentados as matrizes de confusão resultantes da execução dos classificadores *Random Forest* e *Nearest Centroid* executados com os diferentes tipos e abordagens de descritores e com conjunto de treino correspondente a 75% do total de amostras e de teste referente aos outros 25%.

7.1. Resultados no *Random Forest*

Após a execução do algoritmo classificador *Random Forest* obteve-se quatro matrizes de confusão, que são apresentadas a seguir:

13	1	0	0
1	3	0	0
0	0	2	0
0	1	0	21

Resultados: Acertos 39 - Erros 3

Figura 3: Matriz de confusão *Random Forest* - Blocos aleatórios com LPQ

13	1	0	0
1	3	0	0
0	0	2	0
0	1	0	21

Resultados: Acertos 39 - Erros 3

Figura 4: Matriz de confusão *Random Forest* - Blocos fixos com LPQ

13	0	1	0
0	4	0	0
0	0	0	0
0	1	1	21

Resultados: Acertos 39 - Erros 3

Figura 5: Matriz de confusão *Random Forest* - Blocos aleatórios com LBP

13	0	1	0
0	4	0	0
1	0	0	0
0	1	1	21

Resultados: Acertos 38 - Erros 4

Figura 6: Matriz de confusão *Random Forest* - Blocos fixos com LBP

Utilizando o *Random Forest* com o extrator LPQ, tanto para blocos aleatórios quanto para blocos fixos obtivemos a mesma taxa de acerto de 93%. Com o extrator LBP, para os blocos aleatórios também obteve-se uma taxa de 93% de acerto e com os blocos fixos 90% de acerto.

7.2. Resultados do *Nearest Centroid*

Após a execução do algoritmo classificador *Nearest Centroid* obteve-se quatro matrizes de confusão, que são apresentadas a seguir:

11	3	0	1
2	1	0	1
1	1	0	2
0	0	2	17

Resultados: Acertos 29 - Erros 13

Figura 7: Matriz de confusão *Nearest Centroid* - Blocos aleatórios com LPQ

10	4	0	2
2	0	0	0
2	1	0	2
0	0	2	17

Resultados: Acertos 27 - Erros 15

Figura 8: Matriz de confusão *Nearest Centroid* - Blocos fixos com LPQ

13	3	0	2
0	1	0	0
1	1	1	2
0	0	1	17

Resultados: Acertos 32 - Erros 10

Figura 9: Matriz de confusão *Nearest Centroid* - Blocos aleatórios com LBP

10	3	0	1
2	1	0	0
2	1	1	2
0	0	1	18

Resultados: Acertos 30 - Erros 12

Figura 10: Matriz de confusão *Nearest Centroid* - Blocos fixos com LBP

Utilizando o *Nearest Centroid* com o extrator LPQ e blocos aleatórios obtivemos 69% de acerto e com os blocos fixos obtivemos 64% de acerto. Com o extrator LBP, para os blocos aleatórios obtivemos 76% de acerto e para os blocos fixos 71% de acerto.

8. Conclusão

Como apresentado na seção de resultados, a eficiência do classificador *Random Forest* foi muito superior à do *Nearest Centroid*, principalmente a execução utilizando as entradas geradas pelo extrator LPQ. A diferença na taxa de acerto entre os dois classificadores independentemente das variações de descritores utilizada foi de aproximadamente 17%. Esse dado ressalta a importância da escolha adequada do classificador para diferentes problemas. A variação de descritores não impactou tanto na taxa de acerto quando a a variação dos classificadores. Os problemas que tivemos com a segmentação também demonstram a importância de um bom planejamento de cada umas das etapas executadas no processo de reconhecimento de padrões, já que o comprometimento de qualquer um delas pode prejudicar o processo como um todo.

Referências

- Ahonen, T., Rahtu, E., Ojansivu, V., and Heikkila, J. (2008). Recognition of blurred faces using local phase quantization. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.
- Bertolini, D., Oliveira, L. S., Justino, E., and Sabourin, R. (2013). Texture-based descriptors for writer identification and verification. *Expert Systems with Applications*, 40(6):2069–2080.
- Dartora, A. and Oliveira, L. F. Análise de extratores de características para a classificação de tecidos pulmonares e não pulmonares de imagens de tomografia computadorizada de alta resolução.
- Martins, J. G. Uso de descritores de textura extraídos de glcm para o reconhecimento de padrões em diferentes domínios de aplicação.