

Aplicativo classificador de imagens de notas de real para plataforma android

Lais C. R. da Silva Lopes¹, Mikaely C. Silva¹,
Allan K. Lopes¹, Fabrizzio A. A. de Melo N. Soares²

¹Instituto Federal Goiano – Campus Urutaí – Departamento de Informática
Rodovia Geraldo Nascimento, Km 2,5 – Urutaí – GO – Brasil

²Universidade Federal de Goiás – Instituto de Informática
Campus Samambaia, Bloco IMF 1 – Goiânia – GO – Brasil

{allankardec.ti, laiscandidolopes, mikaelycruciol42}@gmail.com, fabrizzio@inf.ufg.br

Abstract. *This paper proposes a system to classify images of Real notes using the data decorrelation technique known as Principal Component Analysis (PCA) and an Artificial Neural Network (ANN). The PCA was used during the preprocessing of images to reduce the size of the data. The new representation of the images, created by the PCA, served as input to an ANN, the type Multilayer Perceptron (MLP) make the classification. After the trained classifier, the state of the objects that compose are preserved and ported to an application developed for the Android platform. The goal is to assist blind people to identify the brazilian banknotes. The results demonstrated that the system was efficient, reaching a hit percentage of almost 100% for all of the images used.*

Resumo. *Este trabalho propõe um sistema classificador de imagens de notas de Real com o uso da técnica de decorrelação de dados, conhecida como Análise das Componentes Principais (PCA), e uma Rede Neural Artificial (RNA). O PCA foi utilizado durante a fase de pré-processamento das imagens para diminuir a dimensão dos dados. A nova representação das imagens, criada pelo PCA, serviu de entrada para uma RNA, do tipo Multilayer Perceptron (MLP), fazer a classificação. Depois de treinado o classificador, o estado dos objetos que o compõe são preservados e portados para um aplicativo desenvolvido para a plataforma Android. O objetivo é auxiliar pessoas cegas a identificar as cédulas monetárias brasileiras. Os resultados obtidos demonstram que o sistema foi eficiente, atingindo uma porcentagem de acerto de quase 100% para o conjunto de imagens utilizado.*

1. Introdução

O Brasil possui uma população significativa de deficientes visuais. De acordo com o censo realizado em 2010 pelo IBGE¹, cerca de 18,8% da população do Brasil possui algum tipo de dificuldade visual [IBGE 2012]. Embora essa população seja expressiva ainda há uma carência muito grande de adaptações para deficientes visuais que enfrentam uma série de dificuldades quando interagem com o ambiente à sua volta. A quantidade de informação codificada visualmente torna essa interação ainda mais desafiadora. Uma

¹Instituto Brasileiro de Geografia e Estatística

dificuldade muito comum de uma pessoa cega é reconhecer o valor das notas de dinheiro que está segurando.

No Brasil, a moeda corrente é o Real composta pelas notas de R\$ 1, 2, 5, 10, 20, 50 e 100. Ressalta-se que, desde a introdução do Real as notas sofreram diversas substituições. Essas substituições foram por causa da evolução de mecanismos de segurança, introdução de elementos históricos ou culturais e em alguns casos acréscimo de elementos para apoio aos deficientes visuais. A nota de R\$ 1, que atualmente está fora de circulação, de 1994 a 2003 teve 20 versões[Banco Central do Brasil 2013]. Cada substituição contribui para a confusão dos deficientes visuais, que podem levar bastante tempo para se familiarizarem com as novas notas.

Diversos trabalhos vem sendo pesquisados para reconhecimento de notas. Em [Hasanuzzaman et al. 2011] foi pesquisado o reconhecimento de notas de dólar utilizando um modelo baseado em componentes a partir da tecnologia *Speeded Up Robust Features* (SURF). No trabalho de [Toytman and Thambidurai 2011] também foi utilizado o reconhecimento de notas de dólar com descritores SURF, porém, com o objetivo de utilização em dispositivos móveis baseados na plataforma Android. [Hinwood et al. 2006] desenvolveu um equipamento para reconhecimento de notas de dinheiro que internamente possui os dispositivos BlindNoteReader, NoteTeller 2 e outros. No trabalho de [Sevkli et al. 2002] foi realizado o reconhecimento de notas utilizando o algoritmo de aprendizado indutivo RULES-3. Em [Aoba et al. 2003] foram experimentadas as RNA's para reconhecimento de notas de Euro com características extraídas de imagens e infravermelho. [Chae et al. 2009] utilizou informações digitalizadas de notas de dinheiro expostas a ultravioleta e comparou com mecanismos contra falsificação previstos nas notas.

Neste trabalho é proposto um sistema para reconhecimento de notas de Real na plataforma Android [Google 2013] usando PCA e uma RNA do tipo MLP. O objetivo é auxiliar pessoas com algum tipo de dificuldade para enxergar a identificarem as notas da moeda brasileira. Ressalta-se que o sistema proposto neste trabalho possui apenas a habilidade de identificar o valor das notas e não constatar sua legitimidade.

2. Revisão sobre análise das componentes principais

O método de descorrelação dos dados, Principal Component Analysis (PCA)[Hotelling 1933] ou Karhunen-Loève Transform (KLT)[Karhunen 1947] e [Loève 1977] tem como objetivo a obtenção de um pequeno número de combinações lineares, componentes principais, de um conjunto de variáveis que retenham o máximo possível da informação contida nas variáveis originais. As componentes são extraídas na ordem da mais explicativa para a menos explicativa, sendo possível descartar as menos importantes, compactando dessa forma a informação original.

Aplicações com imagens que utilizam a técnica geralmente constroem uma matriz formada por um conjunto de vetores $X = \{X_n R^d \mid n = 1, \dots, N\}$ em que as linhas de X são formadas pelos dados de imagens concatenadas, com d sendo o produto entre a altura e largura de uma imagem. Em seguida, calcula-se o vetor média de X conforme equação 1.

$$E(X) = \frac{1}{N} \sum_{n=1}^N X_n \quad (1)$$

O vetor média do conjunto representa tudo aquilo que é comum a todas as imagens. Após subtrair cada elemento de X da média obtém-se uma nova matriz que contém somente a variação de cada imagem em torno da média, para isto cada linha de X deve ser subtraída de $E(X)$, conforme equação 2.

$$\bar{X}_n = \{X_n - E(X), n = 1, \dots, N\} \quad (2)$$

Calcula-se a matriz de covariância $M = \bar{X} \times \bar{X}^T$, encontram-se os autovetores Q e autovalores V da matriz M , para então, recalculamos a matriz de autovetores que seja ortogonal e ortonormal, ou seja, as componentes principais, conforme equação 3.

$$Q^* = \frac{\bar{X}^T \times Q \times (V)^{-0.5}}{\sqrt{N-1}} \quad (3)$$

Finalmente, projeta-se a matriz \bar{X} na matriz transformação Q^* para se obter uma nova representação das imagens. Pode-se ordenar os autovetores da matriz Q^* de acordo com a significância dada pela matriz de autovalores V e assim eliminar os autovetores de menor representatividade.

Para aplicar o PCA em uma imagem de teste, afim de classificá-la, é necessário projetá-la na matriz transformação Q^* . Seja x_{teste} a matriz que representa a imagem de teste; então x_{teste}^T será essa matriz concatenada e \bar{x}_{teste}^T será essa matriz concatenada subtraída da média global $E(x)$, conforme equação 4.

$$\bar{x}_{teste}^T = x_{teste}^T - E(x) \quad (4)$$

Projeta-se \bar{x}_{teste}^T na matriz transformação, conforme equação 5, e utiliza-se uma regra de decisão afim de se classificar a imagem de teste.

$$\bar{x}_{pca} = \bar{x}_{teste}^T \times Q^* \quad (5)$$

3. Modelo proposto

Serão detalhadas a seguir as principais etapas de construção do classificador de notas de Real usando PCA e RNA.

3.1. Digitalização das notas para treinamento e teste

Para o treinamento e teste do sistema foram digitalizadas diversas notas de Real. Foram digitalizadas notas de 2, 5, 10, 20, 50 e 100 reais. As notas de 100 reais foram utilizadas em suas duas versões existentes. As notas foram digitalizadas em frente e verso sob um fundo branco. De acordo com as condições de conservação e semelhança de cada nota, algumas notas não tiveram ambas as faces digitalizadas.

Para a captura das imagens das notas foram utilizadas as câmeras RGB primárias dos celulares Samsung modelos Galaxy Y GT S6102B e Galaxy Pocket S5300, com 3 e 2 megapixels de resolução, respectivamente. As digitalizações foram realizadas manualmente e em diferentes condições de luminosidade, distância e posição. A figura 1 mostra exemplos das imagens de notas de Real capturadas e a tabela 1 apresenta a quantidade de imagens digitalizadas de cada nota.

Tabela 1. Quantitativo de amostras digitalizadas de cada nota

Nota	Qtd. Frente	Qtd. Verso	Total
2	38	38	76
5	35	27	62
10	24	34	58
20	31	32	63
50	35	42	77
100 v1	41	47	88
100 v2	35	41	76
Total	239	261	500



Figura 1. Exemplos de imagens de frente e verso das notas de 2, 10 e 50

3.2. Pré-processamento

Esta fase pode ser dividida em dois passos, no primeiro as imagens obtidas através da atividade descrita na subseção 3.1, foram convertidas em imagens de 256 tons de cinza e redimensionadas para a resolução de 300×140 pixels de largura e altura, respectivamente. Em seguida, dividiu-se as imagens em 2 conjuntos, um para treinamento composto por 90% das imagens e outro para teste com os 10% restantes. Esses conjuntos contêm representantes de todas as classes, por exemplo, para a nota de 2 foram capturadas 38 imagens da parte da frente e 38 do verso, neste caso, separou-se 90% das imagens da parte da frente e do verso para treinamento e 10% das imagens da parte da frente e do verso para teste.

O segundo passo refere-se à aplicação do PCA, descrito na seção 2, ao conjunto de treinamento, o resultado da aplicação do PCA foi utilizado como entrada para a RNA.

3.3. Arquitetura da rede neural artificial

A classificação das imagens foi feita por uma RNA, do tipo *Multilayer Perceptron*, treinada com o algoritmo *backpropagation* [Rumelhart et al. 1986], com *momentum* e taxa de aprendizagem adaptativa [Vogl et al. 1988], para a atualização dos pesos e bias dos neurônios da rede.

A RNA foi projetada com três camadas: uma de entrada, uma oculta e uma de saída. O número de neurônios na camada de entrada é igual ao tamanho dos vetores de entrada que representam as imagens. Essa camada não realiza nenhum processamento,

tem a função de apenas distribuir os sinais para a próxima camada de neurônios. Na camada oculta o número de neurônios, no experimento realizado, variou entre 3 e 60 e para a camada de saída foram necessários 6 neurônios, um por nota.

Os pesos e bias das conexões dos neurônios da rede foram inicializados de acordo com a regra de D. Nguyen e B. Widrow[Nguyen and Widrow 1990]. A figura 2 ilustra a RNA projetada para classificar as imagens. Em ambas as camadas, oculta e de saída, os neurônios utilizam a função de transferência tangente hiperbólica sigmóide.

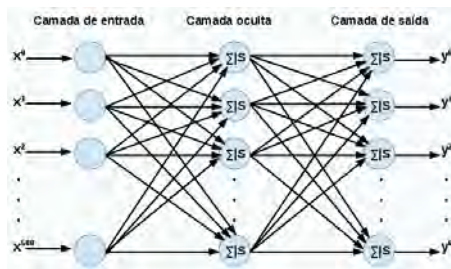


Figura 2. Representação da rede neural projetada

O classificador proposto foi codificado na linguagem de programação Java utilizando o paradigma da orientação a objetos. A figura 3 mostra as principais classes desenvolvidas. Afim de não poluir visualmente o diagrama somente os principais atributos e métodos são mostrados.



Figura 3. Principais classes do sistema proposto

3.4. Treinamento e teste da rna

Como explicado na subseção 3.2, as imagens capturadas foram divididas em dois conjuntos, o primeiro, utilizado durante o treinamento da RNA, composto por 90% das imagens e o segundo, utilizado durante a fase de teste, composto pelos os 10% restantes.

O algoritmo de treinamento da RNA foi executado 20 vezes com diferentes números de neurônios na camada oculta, adotou-se a estratégia de aumentar de 3 em 3 a quantidade de neurônios em cada execução, a primeira execução inicia-se com 3 neurônios. A tabela 2 mostra os parâmetros utilizados nas execuções do algoritmo de treinamento.

Antes do início do aprendizado da RNA aplicou-se o PCA ao conjunto de imagens de treino, conforme o que foi mencionado na seção 2 e na subseção 3.2. Após o treino

empregou-se as equações 4 e 5 a cada uma das imagens do conjunto de teste afim de verificar o desempenho da RNA.

Dois critérios de parada foram adotados nas sessões de treinamento da RNA. O treinamento termina caso o erro mínimo aceitável ou o número máximo de ciclos sejam atingidos. A fim de identificar a causa do término do treino foi adicionado um código para informar a causa de parada (1 - parada por atingir o número máximo de ciclos e 2 - parada por ter atingido o erro mínimo aceitável), conforme pode ser visto na tabela 3.

Tabela 2. Valores dos parâmetros utilizados nas execuções do algoritmo de treinamento da RNA

Parâmetro	Valor	Descrição
lr	0,01	Taxa de aprendizado inicial
inc	1,05	Proporção de incremento da taxa de aprendizado
dec	0,7	Proporção de decremento da taxa de aprendizado
momentum	0,9	Constante <i>momentum</i>
max_inc_erro	1,04	Valor máximo aceitável de aumento do erro médio quadrático
gradiente	0,0000001	Valor do erro global aceitável da rede
epocas	3000	Número máximo de ciclos de treinamento da RNA

3.5. Detalhes da Implementação do Aplicativo para Android

Após o treinamento e teste da RNA, em um computador *desktop*, momento em que teoricamente o classificador estaria pronto para uso, os objetos das classes, mostradas na figura 3, são salvos em disco e reutilizados posteriormente em uma aplicação desenvolvida para Android sem a necessidade de execução de todos os outros passos.

A aplicação foi desenvolvida utilizando o ambiente de desenvolvimento integrado Eclipse versão 4.2.1 codinome Juno, juntamente com o plugin *Android Development Tools* (ADT). Os objetos salvos em disco foram copiados para a memória permanente de um aparelho celular modelo Galaxy Samsung SII GT-I9100. A aplicação desenvolvida realiza as seguintes operações: 1^o - captura a imagem de teste através da câmera do aparelho, 2^o - converte a imagem em 256 tons de cinza e a redimensiona para a resolução de 300×140 de largura e altura, 3^o - aplica o PCA à imagem obtida, conforme equações 4 e 5, 4^o - carrega para a memória principal os objetos armazenados em sua memória permanente e 5^o - classifica a imagem utilizando a RNA.

Para facilitar a utilização do aplicativo foi disponibilizado um atalho na área de trabalho do dispositivo. Uma vez iniciado o programa, o usuário deve apenas direcionar a câmera para a nota, clicar no único botão disponível na interface e esperar pela resposta.

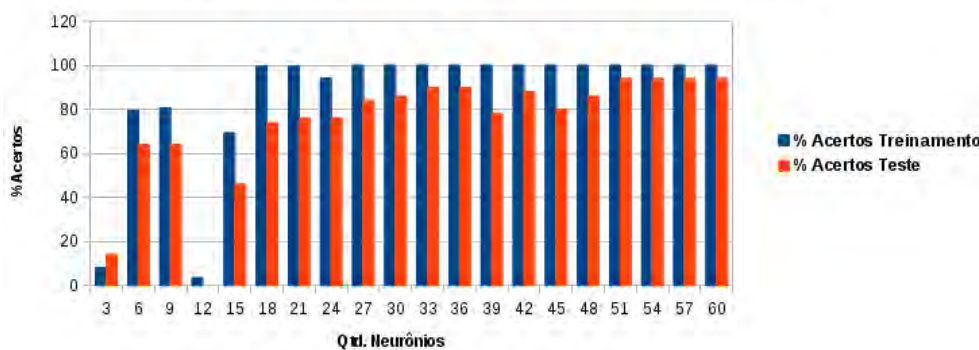
4. Resultados

Na tabela 3 observa-se que a partir de 18 neurônios na camada oculta a RNA já é capaz de fazer uma boa classificação das imagens e que a maioria das execuções do algoritmo de treinamento terminou antes da quantidade máxima de ciclos estipulada (3000), ou seja, convergindo antes para o erro mínimo aceitável de 0,0000001.

O gráfico da figura 4 mostra que a partir de uma certa quantidade de neurônios na camada oculta o desempenho da RNA torna-se estável, não influenciando muito aumentar o número de neurônios nessa camada. Os melhores resultados foram obtidos nas execuções 17, 18, 19 e 20. Nestes casos a RNA errou apenas 3 das 50 amostras do conjunto de teste. Acredita-se que estes resultados podem ser incrementados, ajustando os parâmetros do algoritmo para que convirja mais lentamente. Na maioria das execuções o algoritmo convergiu antes dos 1500 ciclos.

Tabela 3. Porcentagem de acertos da RNA para os conjuntos de treinamento e teste em cada execução

Exec.	Cód. de Parada	Qtd. Neurônios	% Acertos Treinamento	% Acertos Teste
1	1	3	8	14
2	1	6	79,5	64
3	1	9	80,6	64
4	2	12	3,5	0
5	1	15	69,3	46
6	1	18	99,7	74
7	1	21	99,7	76
8	1	24	94,2	76
9	2	27	100	84
10	2	30	100	86
11	2	33	100	90
12	2	36	100	90
13	2	39	100	78
14	2	42	100	88
15	2	45	100	80
16	2	48	100	86
17	2	51	100	94
18	2	54	100	94
19	2	57	100	94
20	2	60	100	94

Relação entre a quantidade de neurônios e o desempenho da RNA**Figura 4. Quantidade de neurônios e o desempenho da RNA**

5. Conclusão

O PCA foi eficiente na redução e caracterização das imagens de notas Real compactando significativamente o tamanho das imagens. Apesar de perda de informação envolvida no processo de obtenção das componentes principais a RNA conseguiu fazer um bom trabalho, aprendendo e generalizando bem os padrões apresentados. Talvez seja necessário modificações no algoritmo de treinamento para que o seu desempenho seja ainda melhor, já que 1500 ciclos de treinamento, em média, talvez não seja o suficiente para a RNA classificar corretamente todas as imagens.

Apesar dos resultados serem positivos é necessário cautela pois as imagens, de certa forma, foram capturadas em um ambiente controlado, todas estão na frente de um fundo branco. Pretende-se capturar outras imagens em ambientes mais reais para apresentar para o classificador já treinado afim de observar como ele se comporta, caso se saia mal, pode-se treinar a RNA novamente.

Em trabalhos futuros pretende-se adicionar novas funcionalidades ao aplicativo desenvolvido afim de torná-lo ainda mais fácil de se utilizar além da realização de estudos comparativos entre o classificador apresentado e outras técnicas de classificação de padrões com o objetivo de desenvolver algo ainda mais eficiente.

References

- Aoba, M., Kikushi, T., and Takefuji, Y. (2003). Euro Banknote Recognition System Using a Three-layered perceptron and RBF Networks. *IPJS Transactions on Mathematical Modeling and Its Application*, 44(SIG 7 (TOM 8)).
- Banco Central do Brasil (2013). Museu de Valores do Banco Central do Brasil.
- Chae, S.-H., Kim, J., and Pan, S. (2009). A Study on the Korean Banknote Recognition Using RGB and UV Information. In Ślęzak, D., Kim, T.-h., Chang, A.-C., Vasilakos, T., Li, M., and Sakurai, K., editors, *Communication and Networking*, volume 56 of *Communications in Computer and Information Science*, pages 477–484. Springer Berlin Heidelberg.
- Google (2013). Sítio oficial do sistema operacional android. www.android.com.
- Hasanuzzaman, F. M., Xiaodong, Y., and YingLi, T. (2011). Robust and effective component-based banknote recognition by SURF features. In *Wireless and Optical Communications Conference (WOCC), 2011 20th Annual*, pages 1–6.
- Hinwood, A., Preston, P., Suaning, G. J., and Lovell, N. H. (2006). Bank note recognition for the vision impaired. *Australasian Physical & Engineering Sciences in Medicine*, 29(2).
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol. Columbia University*, 24:417–441 e 498–520.
- IBGE (2012). Censo 2010. <http://www.censo2010.ibge.gov.br/resultados>.
- Karhunen, K. (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Suomalaisen Tiedeakatemia toimituksia: Mathematica-physica. Universitat Helsinki.
- Loève, M. (1977). *Probability Theory I*. Number 9780387902104 in Graduate Texts in Mathematics. Springer.
- Nguyen, D. and Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, volume 3, pages 21–26. IEEE.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. *Mit Press*.
- Sevкли, M., Türkyilmaz, A., and Aksoy, M. S. (2002). Banknote recognition using inductive learning. Proceedings of Int. Conf. On Fuzzy Syst. And Soft Computational Intelligence in Management and Industrial Eng - 15. FSSCIMIE'02, pages 122–128.
- Toytman, I. and Thambidurai, J. (2011). Banknote recognition on Android platform.
- Vogl, T. P., Mangis, J. K., Rigler, J. K., Zink, W. T., and Alkon, D. L. (1988). Accelerating the convergence of the backpropagation method. *Biological Cybernetics*, 59:257–263.